APPLICATION OF GENERIC SENSE CLASSES IN WORD SENSE DISAMBIGUATION

UPALI SATHYAJITH KOHOMBAN

NATIONAL UNIVERSITY OF SINGAPORE

2006

APPLICATION OF GENERIC SENSE CLASSES IN WORD SENSE DISAMBIGUATION

UPALI SATHYAJITH KOHOMBAN
(B.Sc. Eng(Hons.), SL)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE
2006

Acknowledgements

I am deeply thankful to my supervisor, Dr Lee Wee Sun, for his generous support and guidance, limitless patience, and kind supervision without which this thesis would not have been possible. Much of my research experience and knowledge is due to his unreserved help.

Many thanks to my thesis committee, Professor Chua Tat-Seng and Dr Ng Hwee Tou, for their valuable advice and investment of time, throughout the four years. This work profited much from their valuable comments, teaching and domain knowledge.

Thanks to Dr Kan Min-Yen for his kind support and feedback. Thanks go to Professor Krzysztof Apt for inspiring discussions; and Dr Su Jian, for useful comments.

I'm indebted to Dr Rada Mihalcea, and Dr Ted Pedersen for their interactions and prompt answers for queries. Thanks to Dr Mihalcea for maintaining SENSEVAL data, and Dr Pedersen and his team for the WordNet::Similarity code. I'm thankful to Dr Adam Kilgarriff and Bart Decadt for making available valuable information.

Thanks to my colleagues at the Computational Linguistics lab, Jiang Zheng Ping, Pham Thanh Phong, Chan Yee Seng, Zhao Shanheng, Hendra Setiawan, and Lu Wei for insightful discussions and wonderful time.

I'm grateful to Ms Loo Line Fong and Ms Lou Hui Chu for all the support in the administrative work. They made my life simple.

Thanks to my friends in Singapore, Sri Lanka and elsewhere, whose support is much valued, for being there when needed.

Thanks to my parents and family for their support throughout these years. Words on paper are simply not enough to express my appreciation.

Contents

Summary

Determining the sense of a word within a given context, known as Word Sense Disambiguation (WSD), is a problem in natural language processing, with considerable practical constraints. One of these is the long standing issue of *Knowledge Acquisition Bottleneck* - the practical difficulty of acquiring adequate amounts of learning data. Recent results in WSD show that systems based on supervised learning far outperform those that employ unsupervised learning techniques, stressing the need for labeled data. On the other hand, it has been widely questioned whether the classic 'lexical sample' approach to WSD, which assumes large amounts of labeled training data for each individual word, is scalable for large-scale unrestricted WSD.

In this dissertation, we propose an alternative approach: using generic word sense classes, generic in the sense that they are common among different words. This enables sharing sense information among words, thus allowing reuse of limited amounts of available data, and helping ease the knowledge acquisition bottleneck. These sense classes are coarser grained, and will not necessarily capture finer nuances in word-specific senses. We show that this reduction of granularity is not a problem in itself, as we can capture practically reasonable levels of information within this framework, while reducing the level of complexity found in a contemporary WSD lexicon, such as WORDNET.

Presentation of this idea includes a generalized framework that can use an arbitrary set of generic sense classes, and a mapping of a fine grained lexicon onto these classes. In order to handle large amounts of noisy information due to the diversity of examples, a semantic similarity based technique is introduced that works at the classifier level.

Empirical results show that this framework can use WORDNET lexicographer files (LF) as generic sense classes, with performance levels that rival state-of-the-art in recent SENSEVAL English all-words task evaluation data. However, manual sense classifications such as LFs are not designed to function as classes learnable in a machine learning task; we discuss various issues that can limit their practical performance, and introduce a new scheme of classes among word senses, based on features found within text alone. These classes are neither derived from, nor depend upon any explicit linguistic or semantic theory; they are merely an answer to a practical, end-task oriented, machine learning problem: how to achieve best classifier accuracy from given set of information. Instead of the common approach of optimizing the classifier, our method works by redefining the set of classes so that they form cohesive units in terms of lexical and syntactic features of text. To this end, we introduce several heuristics that modify k-means clustering algorithm to form a set of classes that are more cohesive in terms of features. The resulting classes can outperform the WORDNET LFs in our framework, producing results better than those published on SENSEVAL-3 and most of the results in SENSEVAL-2 English all-words tasks.

The classes formed using clustering are still optimized for the whole lexicon — a constraint that has some negative implications, as it can result in clusters that are good in terms of overall quality, but non-optimal for individual words. We show that this shortcoming can be avoided by forming different sets of similarity classes for individual words; this scheme has all the desirable practical properties of the previous framework, while avoiding some undesirable ones. Additionally, it results in better performance than the universal sense class scheme.

List of Tables

List of Figures

The number of facts we human beings know is,
in a certain very pregnant sense, infinite.
— Bar-Hillel 1960
Language and Information

Chapter 1
An Introduction

This thesis deals with Word Sense Disambiguation – a problem in computational linguistics that focuses on meaning of text at the lexical level. Dictionaries provide us with ample evidence that most words in any human language has more than one meaning. Human language understanding entails figuring out which meaning a word has, in a given context. Word Sense Disambiguation (WSD) in computational linguistics addresses this problem of assigning a word its proper meaning, from an enumeration of possible meanings. State of the art shows that supervised learning with labeled training data can achieve reasonable performance in WSD. However, creating enough training data is known to be expensive both in terms of time and effort. It is this problem, commonly referred to as the *Knowledge Acquisition Bottleneck* (Gale, Church, and Yarowsky, 1992), that motivated the work presented in this thesis.

State of the art in WSD has been based on a *Sense Enumerative Lexicon*, or the idea that words come with lists of senses, each list meant for a given individual word. In contrast, we propose generalizing senses across word boundaries, as *sense classes*; this, in theory, enables us to learn these generic word sense classes as common entities for different words. As the proposed generic sense classes are shared among words, we can reuse available labeled training data for different words. This is helpful in addressing the problem of the knowledge acquisition bottleneck.

1.1 Word Sense Disambiguation

By definition, Word Sense Disambiguation (WSD) is the task of identifying the correct sense of a word in a given context.

This definition involves the concept of *sense*. It is of no doubt that word senses exist, or that language is ambiguous at the lexical level. Consider for instance the word *bank* in the two sentences:

a. Peter got a loan from the bank.

b. The trees grow along the bank of the river.

It is obvious that the two meanings are different, the former denoting a financial institution, and the latter, a slope on ground. This type of word-level ambiguity is known as lexical ambiguity or *polysemy*. Different kinds of lexical ambiguities may occur due to different reasons:

As characterized by the famous 'bank model' shown in the above example, words can, seemingly accidentally, carry totally unrelated meanings. This kind of ambiguity is at sometimes called contrastive ambiguity, or more commonly, *homonymy*. Another type of ambiguity, sometimes referred to as *complementary polysemy*, is more subtle, and involves the difference of usage within the same concept —as in

a. Bob discussed the financing proposal with his bank.

b. The bank is located at the heart of the city.

As far as contemporary computational approaches for WSD is concerned, there is almost no practical difference between different types of lexical ambiguities; different types of senses can be adequately handled by enumerating them in a list for each word. This is the most widespread model assumed in the state of the art of WSD, and in most of the available sense inventories and evaluation schemes.[1] We will call this representation a *Sense Enumerative Lexicon* following Pustejovsky (1995, p.29).

[1] Although some popular sense inventories such as WORDNET come with hierarchical organizations, most evaluation schemes such as SENSEVAL do not take the hierarchy into account.

1.1.1 Utility of WSD as an Intermediate Task

WSD is an intermediate task in natural language processing. In other words, the outcome of a WSD system does not have any use by itself, and is thought to help other tasks in NLP, such as information retrieval and machine translation. However, the opinions are divided on this issue.

In the literature, several comprehensive discussions on the potential uses are available. Probably the most widely cited and the most influential ideas on this issue are those of Bar-Hillel (1970), who was under the strong opinion that fully automatic high-quality machine translation requires that the system understand word meanings. However his ideas on the possibility of attaining good performance levels in WSD, as we will discuss in the section 1.1.2, were sceptical at best.

Recent authors who addressed the issue include Resnik and Yarowsky (1997), Kilgarriff (1997c), Ide and Véronis (1998), Wilks and Stevenson (1996), and Ng (1997). There is a general consensus that WSD does not significantly improve the performance of tasks such as Information Retrieval, which was once considered to be a task that would benefit from WSD (Krovets and Croft, 1992; Sanderson, 1994). Several other authors agree with Bar-Hillel on the potential utility of WSD in Machine Translation (Resnik and Yarowsky, 1997); WSD being a "huge problem" in this area (Kilgarriff, 1997c), and is considered to have "slowed the progress of achieving high quality Machine Translation" (Wilks and Stevenson, 1996); According to Cottrell (1989, p.1), sense ambiguity is "perhaps the most important problem" faced by Natural Language Understanding (NLU). Kilgarriff, however, pointed out that the use of WSD in NLU is not much of a promising area (Kilgarriff, 1997a), whereas there are good chances that Lexicography would benefit much from WSD, and WSD from lexicography (Tugwell and Kilgarriff, 2000). He further shows that the usefulness of WSD in Grammatical Parsing is not established. Carpuat and Wu (2005) showed, with empirical results on English and Chinese language data, that WSD does *not* help machine translation; they claimed that it can even reduce the translation performance by interfering with the language model.

In this work we do not try to establish the usefulness of WSD, or lack thereof, in any particular NLP task. We will limit our attention to the problem of WSD in itself, and focus on the performance as measured by standard WSD evaluation exercises (Edmonds and Cotton, 2001; Snyder and Palmer, 2004).

1.1.2 Possibility of Sense Disambiguation

Interestingly, the very possibility of WSD is a matter of much debate. The issue is far from solved; a reason for this is that the problem itself does not have a clear definition. Not only the question what makes a good WSD system, but also what levels of performance are necessary for WSD to be practically useful in any given task, remain without a solid answer. This undecidedness has led to differing opinions and results on the feasibility of practical WSD.

Bar-Hillel made some important observations on treatment of word meanings, although not in the context of WSD, but of machine translation. He strongly believed that meaning can only be established in logic, and that understanding meaning necessarily entails inference and knowledge. This was extended to a point of suggesting that attaining such a system might possibly be computationally infeasible. He said, that "the task of instructing a machine how to translate from one language it does not and will not understand into another language it does not and will not understand" in itself is a challenging one: if the machine translation system "directly or indirectly depends on the machine's ability to understand the text on which it operates, then the machine will simply be unable to make the step, and the whole operation will come to a full stop" (Bar-Hillel, 1970, p.308).

The famous counterexample Bar-Hillel produced in order to demonstrate his idea on this (Bar-Hillel, 1964, Chapter 12) is essentially a WSD issue, although he did not use the exact term *word sense disambiguation*. The example illustrates the amount of knowledge involved in understanding a seemingly simple text:

> *Little John was looking for his toy box. Finally he found it.*
> *The box was in the pen. John was very happy.*

In order to correctly disambiguate the word *pen* in the sentence, one requires 'world knowledge', such as the relative sizes of pens as writing instruments and as enclosures, and the average size of what can be a toy box, not to mention the physical constraint that an item cannot be placed inside another, if the latter is smaller than the former. The fact that *little John* and *toy box* signals for a child and a play pen is likely to be in the scene, also helps in correctly disambiguating the word *pen*. None of these are available from the text itself, but from world knowledge and requires some inference.

The opinions of Kilgarriff (1997b; 1993) on WSD are mostly based on lexicography rather than on inferential infeasibility. His central argument is that word senses exist only with regard to a particular task, on a particular corpus, and the idea of a universally applicable set of senses "is at odds with theoretical work on the lexicon" (Kilgarriff, 1997b). In particular, he argues that traditional lexicographic artifacts — dictionaries — are prepared for different human audiences and for various uses, and that there is no basis for the assumption that a particular set of senses would suit any given NLP application.

Wilks (1997), addressing the points made by Kilgarriff (1993), admits the possibility that word instances in any corpus can have senses that fall outside any given lexicon, however suggests that this fact alone does not imply a problem, as it may be consistent with the fact that such senses may occupy only an insignificant portion of the corpus. He further suggests that Kilgarriffs idea of corpus based lexicon may be made possible with statistical clustering, though not without practical problems. Our work addresses some of these points.

Some early computational approaches reported results which seemed to suggest that high precision WSD is practically possible and easy. (Yarowsky, 1992) and (Yarowsky, 1995) both reported above 90% accuracy in categorizing words into coarse-grained senses, relying on typically small amounts of manually labeled data. However, this trend did not continue. Possible reasons may be the facts that the level of granularity assumed for senses was too coarse for practical tasks, and that the methods presented were not applicable in general for all words (Wilks, 1997).

Wilks himself claimed that automatic sense tagging of text is possible at high accu-

racy and with less computational effort than has been believed. (Wilks and Stevenson, 1998). He reported that 92% of some 1700-word sample could be disambiguated to homograph level using part of speech alone. The sample they used for evaluation, unlike those of Yarowsky, was unrestricted in the sense that test words were not manually chosen: all open class words from five articles of the Wall Street Journal were used in evaluation. The homograph level selected, from *Longman Dictionary of Contemporary English* (Procter, 1978) could have been coarse, as they also reported that 43% of all open class words in the sample and 88% of all words in the dictionary were monohomographic.[2]

In this work, we do not wish to address the question of theoretical feasibility of WSD; this problem is an issue WSD researchers face as a community at large, and is out of the scope of the matters we deal with. In particular we do not counter the argument of Kilgarriff regarding the impossibility in general of WSD, which is based on theoretical work on lexicography.[3]

1.1.3 The *Status Quo*

The state of the art in unrestricted WSD seems to have somewhat stabilized in terms of both techniques and performance figures. The latter is mostly due to the availability of standard training data, most importantly those of SENSEVAL evaluation exercises (Edmonds and Cotton, 2001; Snyder and Palmer, 2004), which is the result of an effort to standardize WSD evaluation. Another factor is the introduction of WORDNET (Fellbaum, 1998a) and widespread acceptance of WORDNET senses[4] for WSD. Most recently published WSD related work employ WORDNET senses, and most of the available labeled training data is tagged with respect to the same.

Both factors facilitated convenient comparison of different systems, and made it possible to identify which kind of systems generally perform better. Unfortunately, and despite the fact that ideas have been converging, it is still not well known which

[2] Wilks mentioned later (Wilks, 1998) that this claim was "widely misunderstood", although not specifically in which context.

[3] However, the practical implications of this problem cannot be easily brushed off. We will return to this matter in more detail in section 1.3.1.

[4] All our experiments use WORDNET version 1.7.1, unless otherwise specified.

factors necessarily make the best WSD system.

SENSEVAL basically consists of two different types of evaluation, called *Lexical Sample Task* and *All Words Task*. In the lexical sample task, only a selected set of words are tested, and labeled training data is provided. This facilitates a reasonable comparison of the performance of machine learning system alone. All words task, as the name suggests, includes a few documents, and the systems are expected to disambiguate every open-class word in the text. Training data is not provided, and the systems that use supervised learning use whatever the data commonly available.

The accuracy of the best systems in the lexical sample compare well with the agreement levels of human annotators. For instance, the agreement of the first two human annotators in SENSEVAL-3 English lexical sample task was 67.3%, and the best performing system reported an accuracy of 73.9% (Mihalcea, Chklovski, and Kilgarriff, 2004). For the all-words task, the inter-annotator agreement was approximately 72.5%, while the accuracy of the best-performer was only 65.2% (Snyder and Palmer, 2004).

In our opinion, this difference of performance outlines one significant issue regarding the state of the WSD research: machine learning algorithms are already performing satisfactorily when enough training data is available for learning; so the scope of improvement in terms of learning algorithms alone is not very large. On the other hand, the difference in performance between two tasks shows that the techniques that perform well in the lexical sample task do not scale well for unrestricted WSD, which generally lacks enough training data. These two tasks clearly face different challenges: in the lexical sample task, the challenge is how to optimize classifying process assuming enough training data is available; in the all-words task, the most pressing question is how to scale-up WSD for unrestricted text.

This observation is not an isolated one; Wilks noted at a much earlier stage of SENSEVAL that "there is no firm evidence that small scale will scale up to large [scale WSD]" (Wilks, 1998). Some similar ideas were brought up in the SENSEVAL-3 evaluation exercise itself. In the panel on '*Planning* SENSEVAL-4', Lluís Màrquez pointed out the fact that "No substantially new algorithms have been presented" during SENSEVAL-3, and suggested designing new tasks that focus on reusing resources and using available re-

sources (Màrquez, 2004). The non-scalable nature of Lexical Sample task was pointed out by several participants (Mihalcea et al., 2004).

One notable issue is that some systems that performed well in the SENSEVAL all-words task differed from the conventional model of human-annotated data for each individual word by directly or indirectly using clues from related or similar words (Mihalcea, 2002; Mihalcea and Faruque, 2004). These results suggest that there are alternative strategies which can be used for cases where 'conventional' data is not available.

These issues partially motivated us in our topic: generalizing word senses across word boundaries and learn them as general concepts rather than individual word specific senses.

1.2 Argument

The status of the affairs we described above shows that unrestricted WSD is still an unresolved problem, and the major hurdle for solving it is the knowledge acquisition bottleneck, or difficulty in acquiring adequate amounts of training data.

The value of high-quality, expert-annotated labeled training data for WSD cannot be underestimated; however, the reality shows that the acquisition thereof is not practical in terms of time or effort. (We will discuss shortly, in section 1.3, the underlying problems in more detail.) It is this issue that motivated us in this endeavor: to find out ways that generalize the knowledge acquired as much as possible, so the *utility of the limited amounts of already available labeled training data is maximized*.

Our approach for this is based on learning generic sense *classes*. Unlike an enumeration of senses defined for individual fine-grained word senses, these classes can be coarse grained, and they share meanings (and contextual features) among words. The former factor makes learning them easy because the number of classes is reduced, thus increasing the number of training instances per class; the latter helps increasing the amount of training data by making it possible to use labeled instances from different words to learn a particular class, rather than the classic lexical sample approach which

depends on data labeled for each and every different word.

This argument in itself is not convincing, as the current setting of WSD has been to use fine-grained word senses for quite a long time; any suggestion to do otherwise has to show its strength compared to a fine-grained sense setting. This problem, however, can be easily solved if we have a mapping between fine grained senses and sense classes, which we can use to convert senses back and forth between fine and coarse grains. This setting makes it possible for us to

- use whatever the labeled data available for fine-grained senses as labeled examples for sense classes we propose, and,
- use the outputs from coarse-grained classifier to produce fine-grained sense end results.

In what follows, we explain in detail the idea of learning generic senses classes for the end-task of fine grained WSD. Section 1.3 will provide an outline for generic class learning, and argue why we think alternative approaches for unrestricted WSD are worth given a thought. Section 1.3.2 will build our case using empirical evidence that if we can successfully learn a reasonably coarse-grained set of sense classes with enough accuracy, then we can still obtain adequate levels of accuracy at fine-grained WSD.

1.3 Generic Word Sense Classes: What, Why, and How?

Our study focuses on Generic Sense Classes. In this section, we will briefly explain what we mean by generic sense classes, and then we will bring in a few arguments justifying our focus. Taken as a concept, generalizing senses is not a strange idea in semantics, and may be as old as the concept of meaning itself. In its simplest form, the idea means that we can use concepts instead of sense labels. For instance, if we take the word *crane*, we can find two related senses, *'a machine for lifting heavy objects'* or *'a large wading bird'*. Instead of learning these two senses as sense 1 and 2 of *crane*, a learner can use the concepts themselves as sense labels, such as 'bird sense' and 'machine sense'

of *crane*. Once confronted with these, a second learner will be able to instantly identify which sense the first one is referring to, given that he understands the word and is aware of both senses even from a different dictionary from the first learner's. This is not the case with enumerated senses, at least not unless both dictionaries follow the same criteria on numbering senses, and both users are aware of the criteria as well as the related properties of the respective senses that the criteria apply to (such as the frequency within corpus x).

One immediate additional advantage of this scheme is that some of the features we use in language learning can be generalized for these senses. This is possible because the scheme of senses is actually *descriptive* of underlying objects nature, and because they are *common* among different objects. For instance, since the word *crane* has a 'bird sense' and a 'machine sense', it follows that a given ambiguous instance of *crane* can be either a bird or a machine, and generalization follows: Assume that the context shows that this particular instance of *crane* has feathers. If the learner was aware, from previous experience, of the fact that birds normally have feathers but machines do not, he can use this knowledge to quickly disambiguate the sense, even if he did not have any prior experience with either sort of cranes.

This example is very abstract and simplistic; yet it serves as a demonstration of basic features and advantages of a generic sense system. As the human learner could generalize knowledge from related sense knowledge, it can be thought that a WSD system can make use of training examples from related words. In an unrestricted WSD scenario where available training data is very limited, such a method can help maximize the utility of available training data.

1.3.1 Unrestricted WSD and the Knowledge Acquisition Bottleneck

As mentioned earlier, to assume that large amounts of training data will be available for unrestricted WSD is not very realistic. One reason for this is that the effort required for such an endeavor is quite large: Ng (1997) estimated 16-man years for acquiring a labeled corpus of 3,200 most frequently used English words. Mihalcea and Chklovski (2003) estimated "nothing less than 80 man-years of human annotation work" for creat-

Corpus	Description
SEMCOR Corpus	A subset of brown corpus and a novel, tagged with senses of WORDNET , as a part of building WORDNET semantic concordances. Around 230,000 words, with around 180,000 polysemous words. (Landes, Leacock, and Tengi, 1998; Fellbaum, 1998a)
DSO Corpus	Tagged occurrences of common 121 nouns and 70 verbs in English language. Around 192,800 instances, extracted from Brown corpus and Wall Street Journal, hand tagged with WORDNET 1.5 senses. (Ng and Lee, 1996)
SENSEVAL-2 Lexical Sample	Lexical Sample Task provided a set of training data extracted from BNC-2 and Penn Treebank (Wall Street Journal, Brown Corpus and IBM manuals), tagged with WORDNET 1.7 senses. Some 12,000+ instances of 73 words.
SENSEVAL-3 Lexical Sample	Examples extracted mostly from the British National Corpus (BNC), tagged with the help of volunteer contributors in Open Mind Word Expert project (Mihalcea and Chklovski, 2002). 20 nouns, 32 verbs and 5 adjectives, tagged with WORDNET 1.7.1 and WordSmyth senses. (Mihalcea, Chklovski, and Kilgarriff, 2004)
Line, Hard and Serve	Labelled data for words *line*, *hard*, and *serve*, each with 4000+ examples, tagged with WORDNET senses. (Leacock, Towell, and Voorhees, 1993)
Hector Corpus	Made as a pilot project for the BNC, data for 35 words were released in SENSEVAL-1, around 20,000 words tagged with respect to the senses from Hector dictionary. (Atkins, 1992–93)

Table 1.1: Commonly known labeled training corpora for English Word Sense Disambiguation. Some of these are not publicly available.

ing labeled training data for 20,000 words in common English vocabulary. One problem with the above approach is that it is *brute force*, and does not scale well for changing situations such as different sense inventories. Some problematic issues regarding this approach are merely practical, and some are fundamental; a few of them will be discussed shortly.

Currently available amounts of training data does not meet any of these estimates even closely. Table 1.1 shows a brief account of popular sets of training data for English WSD.

To the best of our knowledge, the professionally labeled corpus with the largest

coverage is DSO corpus (Ng and Lee, 1996). This took roughly a man-year of effort, and covers only 121 English words. *Open Mind Word Expert* (Mihalcea and Chklovski, 2003) is a notable attempt to acquire economically a large amount of sense-labeled data, using volunteer help. Although this method eliminates some practical and financial constraints on creating large labeled corpora, it still suffers from the fundamental issues, such as the question of universal suitability of a fixed sense set (Kilgarriff, 1997b). It is worthwhile to discuss several reasons why merely labeling large amounts of data might not help unrestricted WSD outside 'laboratory conditions' found in the Lexical Sample Task.

Unrealistic Universal Sense Sets

First, as Kilgarriff (1997b) pointed out, there is no guarantee that a given sense set would be applicable to every WSD task. This means which tag set is to be used for labeling is a problem to begin with. Even for a given task,agreed-upon 'finalized' sense sets do not exist. For instance, SENSEVAL-3 English lexical sample task switched the sense set for verbs from WORDNET 1.7.1 to WORDSMYTH,[5] citing poor performance with WORDNET verb senses as the reason (Mihalcea, Chklovski, and Kilgarriff, 2004). This brings out the question of which sense set to use in the tagging task.

Second, a set of senses can change with time even within the same lexicon; the largest professionally created data set for WSD, the DSO corpus (Ng and Lee, 1996), is tagged with WORDNET 1.5 senses (released in 1996). Current version of WORDNET is 2.1, and SENSEVAL-2 exercise used version 1.7 of senses (released in 2001). Figure 1.1 shows the differences of the number of senses for 121 nouns and 70 verbs for which DSO annotating was carried out. It can be seen that the majority of words tend to 'collect' more senses in the new versions; it is not easy to automatically convert instances labeled with old senses into new senses, when the number of senses increases and new senses are added. Although it can be expected that a given lexicon will be stable with time, some variations can always be expected.

[5] WORDSMYTH is available at http://www.wordsmyth.net/

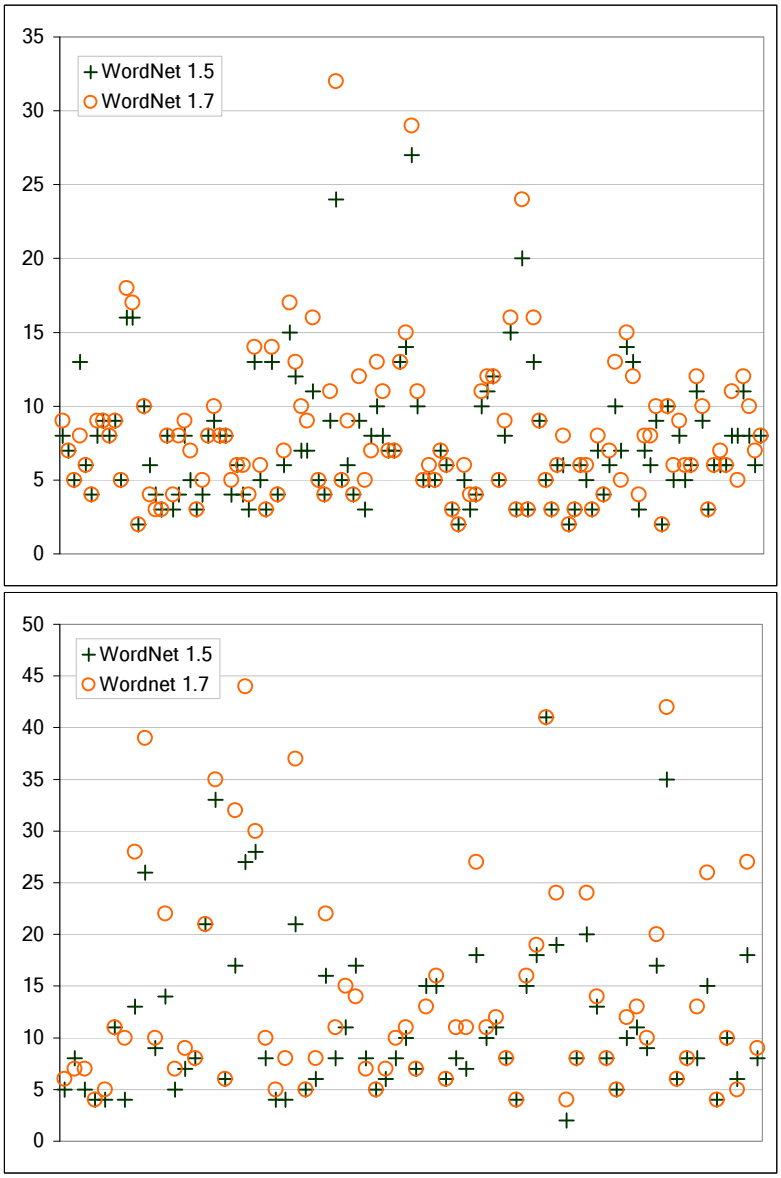

Figure 1.1: Number of senses for 121 nouns (top) and 70 verbs (bottom) used in DSO corpus (Ng and Lee, 1996), in WORDNET 1.5 and WORDNET 1.7. Each point in horizontal axis represents a word, sorted in alphabetical order. Non-overlapping ○ and + points mean a change in the number of senses between versions.

Domain Dependance

The corpus dependence of WSD algorithms' performance is another issue that would make expensive efforts on labeling data questionable on scalability grounds. For instance, experiments of Martinez and Agirre (2000) involved training WSD systems with labeled data from different genres of text, and it was shown that the performance significantly decreases when the training and testing data belong to different genres. Chan and Ng (2005) also claim that WSD systems trained on data from one domain and applied on a different domain show a decrease of performance. Koeling, McCarthy, and Carroll (2005) made similar claims on predominant senses on different corpora.

Same kind of observation was made in the context of the SENSEVAL all-words task by Hoste et al. (2002). They reported that for the words for which they used supervised learning, the overall performance difference between validation data and real test data was nearly 20%.

This would mean that the amount of labeled data actually necessary to handle different texts can be much larger than what is estimated assuming genre independence. Koeling, McCarthy, and Carroll (2005) pointed out that, although the distribution of senses is strongly influenced by the domain, it is not practical to generate labeled data for each domain. Chan and Ng (2006) showed that the sense distribution of the same word in different corpora can be dramatically different. Their example, the noun *interest*, has 6 senses in the DSO corpus (Ng and Lee, 1996). These are senses 1, 2, 3, 4, 5, and 8. In the Brown corpus part of the DSO corpus, these senses occur with the proportions: 34%, 9%, 16%, 14%, 12%, and 15%. The Wall Street Journal part of the same corpus has much different proportions, 13%, 4%, 3%, 56%, 22%, and 2%. (In addition, this provides a good example for the point that was mentioned above in this section: Despite the fact that the sense 8 has 15% in the Brown corpus part —implying that it was considered a significant sense— this sense seems to have been removed in later versions of WORDNET: version 1.7.1 has only 7 senses.)

	nouns		verbs	
	avg. senses	κ value	avg. senses	κ value
before	7.6	0.463	12.8	0.441
after	4.0	0.862	5.6	0.852

Table 1.2: Improvement in the inter-annotator agreement by collapsing fine grained senses into coarser ones, with the reduction of average number of senses (Ng, Lim, and Foo, 1999).

Poor Inter Annotator Agreement on Fine Grained Senses

As discussed with the start of this section, supervised systems that train on substantial amounts of training data is nearing the levels of inter-annotator agreement. This can be thought of as a reasonable upper bound performance level, as attaining higher qualities of inter annotator agreement in labeling data requires greater involvement of the annotators, and will be expensive unavoidably.

It has been observed that the agreement levels can be improved with a coarser set of senses. This is something that can be expected, as human taggers can easily tag more basic senses compared to finer senses that are only slightly different from each other. Ng, Lim, and Foo (1999) provide a quantitative analysis of the effect of coarser set of senses in improving the inter annotator agreement. The experiment involved 121 nouns and 70 verbs frequently used in English, which were labeled in the DSO Corpus project (Ng and Lee, 1996). Their procedure involved a greedy search which collapsed the fine-grained senses into coarse-grained ones aiming to improve the agreement, in terms of κ value, in the process. Table 1.2 shows the improvement levels, along with the reduction of the number of senses, for the words that retained more than one sense after merging senses.

Véronis (1998) also reported somewhat similar results, albeit with smaller improvements. Working on French dictionary senses, he showed that reducing sense distinctions to top-level senses of a dictionary can improve the κ values for nouns, verbs and adjectives respectively from 0.46, 0.41, 0.41 to 0.60, 0.46, 0.46. This might suggest that any system which relies on coarse grained senses by design may be less affected by low-quality annotated data, as it is easier to obtain better agreement on coarse-grained labels.

Some recent approaches for acquiring training data for WSD by less expensive methods, such as (Mihalcea and Chklovski, 2002), employ untrained volunteers' efforts to gather a sizable amount of labeled training data. Annotators who participate in the system can be anyone who is willing to use the interface of the system, which resembles some sort of a word game. As mentioned above, expecting high accuracy for finer-grained sense distinctions from such an exercise would not be very practical, since the taggers involved are not experts in lexicography. Any additional system implemented to verify the inter-annotator agreement will need more effort, and will probably be impractical. Still it can be hoped that they would yield good quality training data for coarse-grained distinctions, because it's easy to agree upon coarse grained senses as we have shown above.

1.3.2 Applicability of Generic Sense Classes in WSD

We do not argue that using large amounts of labeled data is undesirable, or that using generic sense classes is the only way out of the Knowledge Acquisition Bottleneck. Instead, we start from the assumption that available amounts of sense-labeled training data are limited —which is the current reality faced by unrestricted WSD— and propose generalizing senses as *one way* of tackling this issue. The above arguments were meant to show why a new investment on large efforts on labeling data is not guaranteed to conclusively solve the unrestricted WSD problem, hence justifying research on alternative approaches.

On a different note, the problem Mihalcea, Chklovski, and Kilgarriff (2004) mentioned about WORDNET senses' fine granularity merits some attention. Some words in WORDNET have very large number of senses; noun *head* has 32 senses and verb *break* has 63 senses. It can be guessed that this level of sense granularity is a result of an attempt to include every possible usage of a given word in the sense enumeration: this is a problem faced by any sense enumerative lexicon that tries to be complete in coverage. However, it can also be argued that an average user is not conversant with fine nuances, and one will be comfortable with only a few senses for a given word in common usage. One can argue that a coarse-grain set of senses consisting only of these

few senses would reasonably cover all meanings of a word, and would be much easier to handle by a machine learning system.

One can doubt this claim, and oppose the use of coarse-grained senses on the grounds that such senses will *not* adequately cover standard fine-grained senses. We shall address this issue, after a brief description of a model coarse-grained setup.

A Coarse Grained Sense System Based on WORDNET Lexicographer Files

WORDNET senses are organized into *lexicographer files* (LF) by design. LFs provide a rough thematic arrangement, such that the senses which fall into the same LF share a common conceptual theme. For instance, the first senses of words *cat* and *dog* fall into the LF **NOUN.ANIMAL**. All WORDNET noun senses fall into 26 LFs,[6] and verbs into 15; this is a fairly coarse generic mapping. This arrangement will be discussed in detail in section 2.2.3; for now it suffices to say that the LFs provide a convenient method of forming a natural coarse-grained set of senses out of WORDNET fine-grained senses.

This method is to eliminate some of the finer senses of a word by keeping only one sense per LF. For instance, the four senses of *building* in WORDNET are

sense 1 **ARTIFACT**: a permanent structure that has a roof and walls

sense 2 **ACT**: the act of constructing or building something

sense 3 **ACT**: the commercial activity involved in constructing buildings

sense 4 **GROUP**: the occupants of a building

Shown in **SMALL CAPS** are the LFs associated with each sense. It can be seen that senses 2 and 3 are related to each other, and share the same origin of meaning, while senses 1 and 4 have meanings different from this and from each other. It is possible to lump the four senses into three coarser ones: *'physical structure'*, *'act of construction'*, and *'building occupants'*, which adhere to the LF arrangement. A simple heuristic for lumping is to keep the sense with the lowest sense number[7] for each LF and discard

[6] one of these, **NOUN.TOPS**, is a 'maintenance' grouping that does not have a semantic theme.

[7] This is motivated by the fact that WORDNET senses come in descending order of their frequency in a labeled corpus. We shall revisit this matter shortly.

all the rest, and consider any instance that was earlier assigned a discarded sense of a given LF as an instance of the retained sense for the same LF.

As mentioned earlier in this section with the *crane* example, this assignment has an added advantage. It is possible to pose this problem as one of learning LFs instead of senses. Since senses from different words fall into the LFs, it is possible to use examples from different words for learning LFs. For instance, to learn the sense **ARTIFACT** of *building/1*, labeled examples from sense *house/1* can be used, as this sense also falls into the same LF.

With this simple arrangement, it is now possible to address the issue about usefulness of the coverage of coarse-grained senses. Although the above example shows that the number of senses reduced by 25%, the actual frequencies of the senses are very skewed in natural language. For instance, out of 52 labeled instances of *building* in SEMCOR corpus, 48 belongs to sense 1, and the rest occupy 3, 1, and 0 instances respectively. In other words, we lose the exact fine-grained sense for only one instance, out of 52, for this word.

Figure 1.2 shows the total reduction of number of senses and proportion they actually occupy in SEMCOR corpus, for polysemous nouns and verbs.

How Far can Coarse Grained Senses Go?

In order to quickly evaluate the effect of coarse-grain loss in real-world tasks, a small experiment can be conducted with a WSD benchmark - the SENSEVAL English all-words task. Each instance from the official answer key in SENSEVAL tasks was analyzed, and the LF it belongs to was found out. It is straightforward to do this, as the sense to LF mapping in WORDNET can be readily extracted. A list of 'answers' was built out of the official answer keys, by replacing each answer key with the sense that has the smallest sense number within the LF of the original answer key. If this sense is the same as the original sense of the answer, this answer instance is not affected by our switching to a coarse-grained sense set. On the other hand, if the original answer key is something other than the sense with smallest number, it means that the original answer key falls outside the coverage of our coarse-grained sense set. This will introduce an

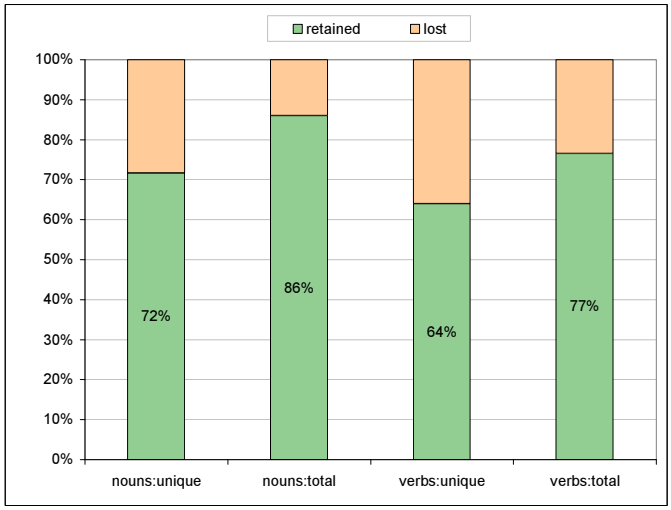

Figure 1.2: Proportions occupied by LFs first and secondary senses for polysemous nouns and verbs in SEMCOR: retained: senses with the smallest sense number within a LF, lost: senses falling into other sense numbers, hence losing their original senses in a coarse-grained mapping. The graph shows the reduction of individual sense count (unique) and total number of instances (total). Proportional loss of actual instances is considerably smaller than the reduction of the number of senses, due to the skewness of sense distribution.

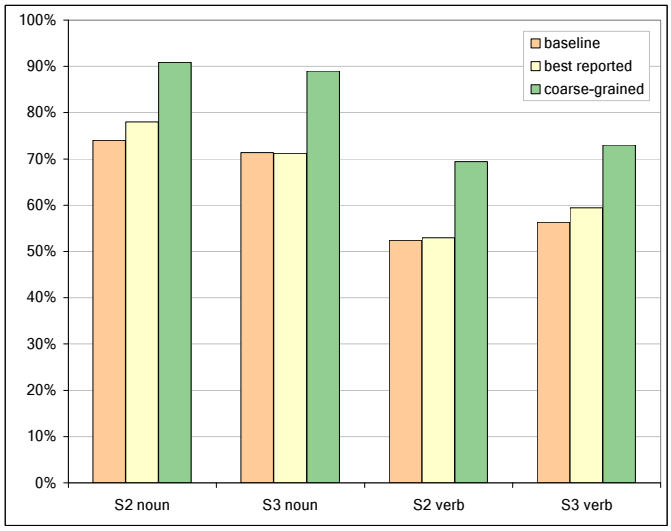

Figure 1.3: SENSEVAL performance of Baseline, best SENSEVAL systems and the upper-bound performance of hypothetical LF-level coarse grained classifier. Given that a system can more easily learn the coarse grained senses, there is a reasonable room left for improvement.

error for that instance during evaluation.

Given this setup, it is straightforward to calculate the proportion of errors induced by discarding a few senses in the way described previously. This will serve as an upper bound of the accuracy of a hypothetical coarse-grained sense classifier, which used only the most frequent sense per LF in SENSEVAL tasks.

Figure 1.3 shows the upper bound performance of this hypothetical classifier in SENSEVAL-2 and 3 all-words tasks, along with the baseline (WORDNET first sense) performance, and the performance of the best reported systems. Since WORDNET LFs are properly defined only for nouns and verbs, performance values for only these two parts of speech are reported.

It can be seen that, compared to the improvement of the best reported systems' performance over baseline, the upper-bound performance of our hypothetical classifier is much higher. This shows that the loss due to coarse-grained senses alone is not a reason to reject a suitably designed coarse-grained system. If it is possible to gain some advantage over conventional senses by using coarse-grained senses, there still is a reasonable room left for improvement.

1.4 Scope and Research Questions

Above section concludes the outline of the basic research problem addressed in this thesis: generalizing word sense knowledge. Generalizing senses is itself a problem with a number of theoretical issues, most with roots that go straight into theoretical linguistics and cognitive science. We do not plan to venture into this area, but confine ourselves to computational aspects of the problem.

In particular, the domain of our main interest is word sense disambiguation, and *unrestricted* setting thereof, where the lack of training data is a fact one has to live with. For practical reasons, we will restrict most of our experiments to the resources publicly available, both for implementation and evaluation. In case of implementation, this will help us to argue that our system is feasible and useful with respect to practical realities; in case of evaluation, this will enable easy comparisons with the state of the art.

We will address the following questions:

- whether generic sense classes can help large-scale, unrestricted, fine-grained WSD
- whether learning generic sense classes is possible with available practical technology, and if it is,
- how we can learn generic sense classes and use them in fine grained WSD, and what kind of practical issues are relevant in generic class learning, and
- how these issues can be addressed in order to improve the effectiveness in learning generic classes.

1.5 Contributions

This thesis finds reasonably favorable answers to most of the above questions.

It is demonstrated, using WORDNET lexicographer files as our generic sense classes, that learning generic sense classes is indeed possible to be done with reasonable accuracy. To this end, a technique is introduced to use semantic similarity between concepts during the classifier process, in order to optimize the classifier output in sparse data conditions. The results obtained in fine-grained unrestricted WSD, on the evaluation data sets of recent SENSEVAL tasks in particular, rival the state of the art.

In addition, several theoretical and practical issues related to learning generic sense classes, using contextual features of text, are identified. Based on these observations, techniques are developed that can create a set of generic classes, which is specifically designed for the end-task at hand —automatic classification using machine learning techniques and automatically acquired contextual features. It is empirically shown that these classes are preferable for fine-grained sense learning, as they increase the granularity of sense divisions. In addition, they can provide better performance in the WSD end task.

The ideas used in the above exercise are concerned mainly on generalizing information within senses. The grouping of senses into 'bins' according to usage has one

remaining practical weakness: it can lead to over-generalizing, introducing unnecessary relationships between senses, because the grouping of senses is only concerned on the quality of the group as a whole. We show that we can avoid this by calculating the similarity-classes per individual word, rather than a common set of sense classes for all words; this yields better performance over generic classes, while keeping the practically useful attributes of the generic sense class framework intact.

1.5.1 Research Outcomes

As mentioned earlier, the work presented in this thesis introduced WSD techniques which could learn generic sense classes from limited amounts of training data, and rival the state of the art systems in fine grained WSD. A number of papers resulting from this work has been published or are in preparation.

- Kohomban, Upali S. and Lee, Wee Sun. *'Learning Semantic Classes for Word Sense Disambiguation'*. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL05), June 2005
- Kohomban, Upali S., Lee, Wee Sun *'Optimizing Classifier Performance In Word Sense Disambiguation By Redefining Sense Classes'* Proceedings of the twentieth International Joint Conference on Artificial Intelligence (IJCAI-07), January 2007.
- Kohomban, U. S. 'Using sense classes in unrestricted WSD' in preparation.

1.6 Chapter Summaries

Chapter 2 will describe an outline of the generic sense class framework we propose. It will articulate the basis for learning generic sense classes as a work-around for some constraints faced by contemporary research on unrestricted WSD. Then it will discuss several theoretical schemes that attempt to generalize word sense knowledge. At the end, it will provide a comprehensive introduction to the framework of learning generic classes for WSD, and introduce some definitions. Finally a discussion is provided on previous related work.

Chapter 3 Describes a proof-of-concept system that is based on the framework introduced in the previous chapter. This system is essentially a generic implementation of the framework; however for illustrative purposes, WORDNET lexicographer files are used as the set of generic sense classes. It will describe the additional technical details pertaining to the implementation.

Chapter 4 presents the performance of the system described in the previous chapter, applied on SEMCOR corpus data and SENSEVAL English all-words task evaluation data. It will also discuss the implications of using WORDNET lexicographer files, and how the performance can be affected by these.

Chapter 5 is a discussion on the use of WORDNET lexicographer files as generic sense classes for WSD. In this chapter, the issues are addressed from a practical end-task perspective, as well as from the point of observations of some theoretical work on lexicons and semantics. The argument is aimed to show that the WORDNET lexicographer files are not designed keeping sense disambiguation in mind, and are associated with practical problematic issues. It concludes with a discussion on the desirable features of a sense class system meant for automatic word sense disambiguation.

Chapter 6 implements an automated sense clustering system that tries to address the arguments raised in chapter 5. It will describe the clustering algorithms employed in order to create a set of generic sense classes that are supposed to work better in our machine learning problem, and yield better results in fine grained WSD. The chapter also presents empirical results of using these classes in place for WORDNET lexicographer files in the framework presented in chapter 3, and argue that they can improve the performance over WORDNET lexicographer files.

Chapter 7 introduces another extension on this; here, a partitioning system is described, which can cluster senses into sense maps defined for individual words instead of a common map for all words. The partitions still retain some form of generic nature as the senses that fall into the same partition can be thought to be in the same class as

the partition center. However, this approach is more flexible than the global clustering scheme, because it is possible to optimize the clusters per individual words, rather than optimizing them for a global minimum of variance.

Again, with the results on SENSEVAL evaluation data, it is shown that the system can improve over globally defined set of clusters.

Chapter 8 concludes the thesis, and discusses some areas that are worth a thought.

1.7 Summary

Word Sense Disambiguation, though not an end task in NLP in itself, has many unsolved practical questions. In this chapter the nature of some of these problems were discussed, along with a description of the state of the art of WSD. Knowledge Acquisition Bottleneck continues to be one of the biggest hurdles for practical unrestricted WSD; it was argued in this chapter that some of the conventional techniques, the classic lexical sample approach in particular, does not hold much promise as far as the scalability is concerned.

Our argument was not to question the conventional approach on performance grounds, but to present an alternative that can help overcome the knowledge acquisition bottleneck. In section 1.3, a coarse grained set of generic classes was proposed as a way of overcoming data scarcity. This method works by

- limiting the number of senses per word to a concept-level, and
- reusing the concept-knowledge among different words.

This setting can reduce the number of senses in the lexicon without excessively compromising the accuracy in real world WSD tasks.

Entia non sunt multiplicanda praeter necessitatem.
Entities shall not be multiplied beyond necessity.
— William of Occam c. 1285-1349

Chapter 2
Senses and Supersenses

Generalizing specific word senses into more generic ones is not uncommon in perhaps any language, as this results from the universal fact that senses usually denote concepts, and concepts themselves can be generalized. Almost any concept can be described as a specific case of a more general concept; for instance, *horse* is an instance of *mammal*, and *mammal* itself is a specific case of *animal*.

The idea of using common usage or thematic patterns for categorizing words into 'classes' was the focus on many research work and theoretical studies. Traditionally, in the context of WSD, the pressing reason for generalizing was based on practical problems —sparsity or lack of training data— which also motivated unsupervised or dictionary based approaches, which benefitted from generalizing at times. However, in the theoretical front, some arguments are focused on the problems inherent to the Sense Enumerative Lexicon. We discussed in section 1.1.2 the opinion of Kilgarriff (1997b) on the universal suitability of a particular set of senses (although he did not propose generalizing as a solution to this problem). Pustejovsky (1995) also identifies several problems with a sense enumerative lexicon, including the inability to cover creative uses of words and the assumption on rigidity of senses, or the assumption that the senses have non-overlapping boundaries. Although not as expressive as his remedy for the problem —a *generative* lexicon— simple generalizing schemes can handle some of these issues in practice by providing some level of abstraction for concept definitions.

In this chapter, we will study a set of generalization schemes that have been presented for WSD and related areas. We will then move on to WORDNET, undoubtedly the most authoritative reference of word senses in recent computational linguistic work in general, and in WSD in particular. We will discuss a few important design features of WORDNET that will facilitate us present our major idea: a classification of fine grained word senses into coarse-grained clusters or *sense classes*, which we suggest learning instead of fine grained word specific *enumeration* of senses. Using WORDNET lexicographer files as the set of generic classes, we demonstrate how we can employ our system for the end goal of fine-grained WSD.

2.1 Generalizing Schemes

Sense generalizing schemes can be divided into two broad categories depending on whether they do or do not assume a particular structure underlying the sense organization, which is known *a priori*. Class based schemes assume such a structure and build upon it, while 'class-free', or similarity based, systems work directly on similarities in corpus distributional properties of different entities; they do not assume, or try to deduce, an explicit structure.

2.1.1 Class Based Schemes

Class based schemes utilize either established or automatically created set of classes, in order to derive the notion of commonality. For instance, the *crane* example we discussed earlier showed that the WORDNET-encoded fact of *crane* being a **BIRD** can be utilized for acquiring data not available explicitly.

Yarowsky (1992) proposed a method of disambiguating word senses using topical categories. The categories he used come from Roget's International Thesaurus. There are 1042 categories in total, and a word can fall into different categories depending on its senses; for instance, the word *crane* can fall into either **MACHINE** or **ANIMAL** category. His method is largely unsupervised. First, a large number of examples are gathered for each category from a corpus, and are used to find contexts which typi-

cally represent a category. The corpus he used was Grolier Encyclopedia. Note that the words are not category-labeled and may actually be examples from a wrong category. However, the frequencies of the 'salient words' representative of a category accumulate for the category, while frequencies of spuriously occurring words are distributed among other 1041 categories. Thus, the 'signal' is concentrated only in the correct category. The measure of 'salience' can be used to identify the category any context represents. Yarowsky defined salience of a word as $\log \frac{P(word|category)}{P(word)}$. Disambiguating a new word instance is straightforward: first the category of the context of the word instance is identified, and then the sense that is associated with the category is assigned for the word. For instance, if the word *crane* appears in a context where words such as *species*, *family*, *bird*, *fish*, and *breed* are also present, then it can be identified as an instance of **ANIMAL** sense, as the companion words are salient for the **ANIMAL** category.

Resnik (1997) provided a method based on a similar intuition, but on syntactic clues rather than broad contextual ones. His method is based on *selectional preferences* (Resnik, 1996; Resnik, 1993), which are essentially syntactic predicates that *select* for a particular class of a word: For instance, the verb *drink* selects for a **BEVERAGE** as its object. Resnik's was an attempt to model common behavior of words after selectional constraints (Katz and Fodor, 1963) using statistical properties from real text. In (Resnik, 1997) he used verb-object, verb-subject, head-modifier, and adjective-noun relationships in order to define the selectional preferences.

This scheme is worth further discussion, as some of the issues related are much relevant to the problems our system is facing as well. Resnik defined the *selectional preference strength*, $S_R(p)$ of a given predicate p as the Kullback-Liebler divergence (Kullback and Leibler, 1951) of $P(c|p)$ from $P(c)$ where C is the set of classes applicable. Thus,

$$\begin{aligned} S_R(p) &= D(P(c|p)\|P(c)) \\ &= \sum_c P(c|p) \log \frac{P(c|p)}{p(c)} \end{aligned}$$

For the predicate `object-of-drink`, the class **BEVERAGE** will have very high probability. Then the term *selectional association* $A_R(p, c)$ of the predicate with a given

class c is defined as the proportion of the component it contributes to the overall selectional preference strength of the predicate. In other words,

$$A_R(p,c) = \frac{1}{S_R(p)} \cdot P(c|p) \log \frac{P(c|p)}{p(c)}.$$

For a given word matching a predicate p, disambiguation is done by finding out which of the classes the word can fall into has best selectional association with p.

Note that this scheme is only concerned about the overall probabilities of classes and the conditional probability of each class given the predicate, without regard to the underlying word to which the predicate applies.

The scheme reported in (Lin, 1997) is similar, but uses an aggregate measure on words that maximizes the likelihood of a word belonging to a particular class. The 'local contexts' Lin defines are not very different from the selectional preferences; This approach of disambiguating senses involves first identifying the 'selector' words that have similar local contexts to the word being disambiguated, and picking the sense that maximizes the similarity between the target word and those selectors. The similarity measure is described later in section 2.3.1, and is based on WORDNET hierarchy.

Mihalcea and Faruque (2004) employed a scheme of using contextual patterns from hierarchically-related words from WORDNET, as clues for word instances for which there is no labeled training data.

Basili, Rocca, and Pazienza (1997) suggested that the class system must be adjusted for the underlying corpus. Their method is able to derive from a corpus a tag set that is suitable for semantic tagging of words in the domain of that particular corpus. The tags are picked from higher level concepts of WORDNET taxonomy.

2.1.2 Similarity Based Schemes

Generalizing does not necessarily need a classification of senses. Information about a word can be derived from 'similar' word instances even when one does not have any labeled relationship or grouping between the two instances. Dagan and colleagues (1993; 1994) discuss how word co-occurrence probabilities can be used to derive in-

formation about unseen cases of words. This method does not assume any classes, and merely depends on the *similarity* between word pairs. This similarity is defined in terms of co-occurrence pairs, or the occurrence of two words within a fixed size window of words; words that have similar co-occurrence patterns with other words are considered similar. However, any kind of relation, for instance Resnik-style selectional association (section 2.1.1) can be employed, as the method does not assume a particular model on similarity. As an example, suppose we want to estimate the mutual information between words *chapter* and *describes*, but do not have any corpus instances with both words. However, if we know that words *introduction*, *book*, and *section* has similar mutual patterns to those of *chapter* with other words, and if we have instances of these three words co-occurring with *describes*, then it is possible to estimate the required value for *chapter–describes*. The authors used a method of averaging values from similar words. This is much similar to the method reported by Lin (1997) we described above, except that Lin's method used WORDNET hierarchy for calculating the similarity measure.

2.2 WORDNET: The Lexical Database

One of the most widely used lexicons in computational linguistics is WORDNET (Fellbaum, 1998a). WORDNET is a lexical database that provides information on *relations between word senses*, in addition to sense definitions or glosses. These relationships make it a rich source of information, with a wide range of applications. Although the creators claim the original intention was "to identify the most important lexical nodes by character strings and to explore the patterns of semantic relations among them" (Fellbaum, 1998a, p. xvii), the popularity of the WORDNET senses as the reference lexicon grew over time, as evident from a large body of WSD research involving WORDNET senses. The human-annotated relationships among word senses played some part in attaining this popularity.

The inter-sense[1] relationships in WORDNET include *synonym/antonym* (senses that

[1] WORDNET senses are grouped in to *synsets*, which are sets of senses of different words with identical

crane *large long-necked wading bird of marshes and plains in many parts of the world*
⇒ **wading bird** *long-legged birds that wade in water in search of food*
⇒ **aquatic bird** *wading and swimming and diving birds*
⇒ **bird** *warm-blooded egg-laying vertebrates characterized by feathers and wings*
⇒ **vertebrate** *animals having a skeleton with a spinal column*
⇒ **chordate** *any animal of the phylum Chordata*
⇒ **animal** *a living organism characterized by voluntary movement*

Figure 2.1: Hypernym hierarchy for noun *crane* ('ANIMAL' sense). Each successive step is more general than the previous one, characterized by 'is-a' semantic relationship.

have similar/opposite meanings), *hypernym/hyponym* (senses that are more generic/specific than the other), *meronym/holonym* (whole/part relationship between concepts), among others. Probably, the most basic kind of generalization relationships is hypernymy; in this work we are mostly interested in this relationship.

2.2.1 Hypernym Hierarchy

Most semantic hierarchies are based on *is-a* relationships, which link senses with more general concepts. Technically, this relationship is called hypernymy, and its opposite hyponymy: for instance, *animal* is a hypernym of *bird*, and *bird* is a hyponym of *animal*.[2]

An *is-a* relationship means semantic inheritance, and hypernyms serve as semantic superclasses for underlying senses. This provides a facility that is not available in a conventional dictionary, as we can use the pointers within the database to derive additional information from the system *automatically*, allowing a convenient way of accessing, and a compact way of storing, taxonomical semantic information. In the example above, the definition of *crane* provides a minimal description, and the rest hierarchically follow; at the next level we know that crane wades in water for food, and two levels higher, that it lays eggs and that it has wings. Any of these information can possibly be used in differentiating an 'ANIMAL' sense instance of *crane* from its other 'MACHINE' sense.

meanings, such as *spouse/1* and *partner/1*. Most of the relationships are in fact defined over synsets, not over senses.

[2]Distinction of different senses of the same word is important in WORDNET and in this case sense 1 of *animal* is a hypernym of sense 1 of *bird*. Also, we use 'hypernym' here in a transitive sense; that is, a concept *a* does not have to be the immediate parent of concept *b* to be called the hypernym of *b*, as long as *a* can be linked to *b* through a consecutive series of hypernym links.

Fellbaum (1998b) suggests that the concept of hypernymy for verbs is not solidly defined in human mind as is the case for nouns. According to her, study of verb hypernyms show evidence for 'many kinds of semantic elaborations across different semantic fields' —for example, conflations of move and semantic components such as MANNER and CLAUSE, exemplified by *slide* and *pull*. WORDNET does not pay much attention to all the conflations, 'since the aim is to study relations between verbs, rather than between the building blocks that make them up', and focuses on manner relationship. The depth of verb hierarchy is rather shallow, not exceeding 4 levels in many cases. Almost every taxonomy of verb has a 'bulge' at some level, which is a concentration of lexicalization; both above and below this level, concepts are not richly distributed (Fellbaum, 1998b).

The conventional lexical sample approach to WSD ignores the fact that 'crane is a bird', and expects explicit labeled instances of *crane* in training data to be present which would indicate that cranes have wings, cranes lay eggs etc. It does not make use of the implicit information from the hierarchy mentioned above; in fact, it does not recognize the semantics of the two meanings at all, relying totally on statistical patterns in labeled examples. In our opinion, this is the weakest point in the lexical sample approach when applied to a sparse-data condition, where such patterns may be absent or too weak. For instance, in the one-million word Brown Corpus (Francis and Kucera, 1982), words *cigarette* and *tobacco* co-occur with word *ash* only once: this is the same frequency that *ash* co-occurs with *room*, *bubble*, and *house* (Ide and Véronis, 1998).

2.2.2 Adjectives and Adverbs

Adjectives are not organized into hypernym-based tree structure. The organization of adjectives in WORDNET is based on the work of Gross, Fischer, and Miller (1989): They suggested that adjectives be better organized into opposite 'poles' that denote semantic opposites, such as *dry* and *wet* in the example in figure 2.2, and 'satellite' adjectives that connect to those poles. The satellite adjectives denote different variants of the basic senses denoted by the poles.

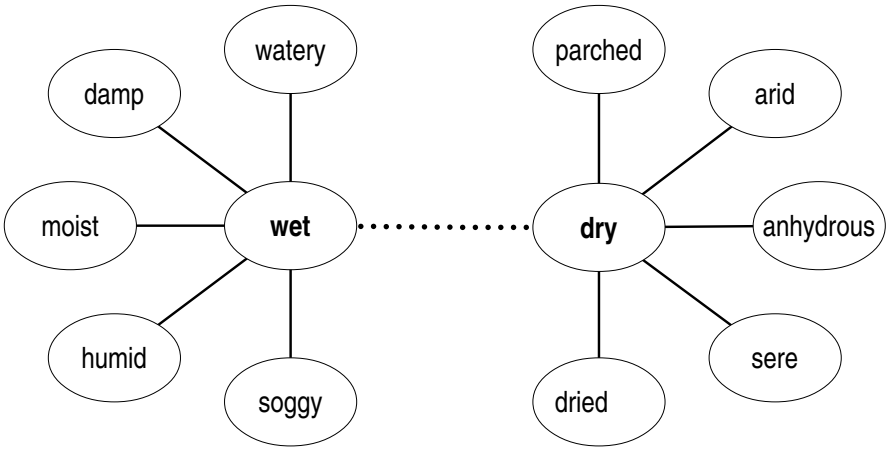

Figure 2.2: Adjective organization in WORDNET

Most adverbs do not need their own classification as they can be easily linked to the adjective they are derived from: for instance *quickly* can link to *quick*. They do not have a tree structure like nouns and verbs, or a two-pole cluster stricture like adjectives.

2.2.3 Lexicographer Files

As we discussed in section 1.3.2, WORDNET lexicographer file (LF) arrangement provides a convenient generic coarse grained class system. The lexicographer files divide nouns and verbs into several hierarchies.

For nouns, there is a related concept known as *Unique Beginners* which is important in the hierarchy. These are related to lexicographer files. Originally, the noun hierarchy was divided into 25 top level concepts. However, it was apparent that some of the concepts could be grouped within others, and these concepts were organized into the hierarchy itself. This resulted in 11 true beginners which did not have any parent. Original 25 remained as lexicographer files, and in addition, a new lexicographer file was created to handle concepts that did not have parents, named **NOUN.TOPS**. (Fellbaum, 1998a, Chapter 1).

Verbs divisions start with one major cut, which separates actions and events from states. The actions and events comprise most words, and these get subdivided into 14 more domains, These 14, together with stative words, form 15 lexicographer files (Fellbaum, 1998a, Chapter 3).

File Number	Name	Contents
00	ADJ.ALL	all adjective clusters
01	ADJ.PERT	relational adjectives (pertainyms)
02	ADV.ALL	all adverbs
03	NOUN.TOPS	unique beginners for nouns
04	NOUN.ACT	acts or actions
05	NOUN.ANIMAL	animals
06	NOUN.ARTIFACT	man-made objects
07	NOUN.ATTRIBUTE	attributes of people and objects
08	NOUN.BODY	body parts
09	NOUN.COGNITION	cognitive processes and contents
10	NOUN.COMMUNICATION	communicative processes and contents
11	NOUN.EVENT	natural events
12	NOUN.FEELING	feelings and emotions
13	NOUN.FOOD	foods and drinks
14	NOUN.GROUP	groupings of people or objects
15	NOUN.LOCATION	spatial position
16	NOUN.MOTIVE	goals
17	NOUN.OBJECT	natural objects (not man-made)
18	NOUN.PERSON	people
19	NOUN.PHENOMENON	natural phenomena
20	NOUN.PLANT	plants
21	NOUN.POSSESSION	possession and transfer of possession
22	NOUN.PROCESS	natural processes
23	NOUN.QUANTITY	quantities and units of measure
24	NOUN.RELATION	relations between people or things or ideas
25	NOUN.SHAPE	two and three dimensional shapes
26	NOUN.STATE	stable states of affairs
27	NOUN.SUBSTANCE	substances
28	NOUN.TIME	time and temporal relations
29	VERB.BODY	grooming, dressing and bodily care
30	VERB.CHANGE	size, temperature change, intensifying, etc.
31	VERB.COGNITION	thinking, judging, analyzing, doubting
32	VERB.COMMUNICATION	telling, asking, ordering, singing
33	VERB.COMPETITION	fighting, athletic activities
34	VERB.CONSUMPTION	eating and drinking
35	VERB.CONTACT	touching, hitting, tying, digging
36	VERB.CREATION	sewing, baking, painting, performing
37	VERB.EMOTION	feeling
38	VERB.MOTION	walking, flying, swimming
39	VERB.PERCEPTION	seeing, hearing, feeling
40	VERB.POSSESSION	buying, selling, owning
41	VERB.SOCIAL	political and social activities and events
42	VERB.STATIVE	being, having, spatial relations
43	VERB.WEATHER	raining, snowing, thawing, thundering
44	ADJ.PPL	participial adjectives

Table 2.1: WORDNET lexicographer files

LFs → senses ↓	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Average num. of LFs**
1	94714										**1.0**
2	3966	5450									**1.6**
3	558	1220	932								**2.1**
4	117	297	400	213							**2.7**
5	36	113	169	150	67						**3.2**
6	11	30	81	89	63	19					**3.8**
7	7	16	36	54	55	18	4				**4.1**
8	2	3	13	27	18	20	5				**4.5**
9		3	5	14	23	19	7	1			**5.0**
10	2	1	4	9	15	9	11	4			**5.3**
11	1	1		5	10	7	5	4	1		**5.6**
12				5	4	1	4	2	3		**6.2**
13		1	1		3	1	2	2	2	1	**6.5**
14							3		1	1	**8.0**
15					2	1	1	2	1		**6.9**
16					1	3		1	1		**6.7**
17						1	3	2			**7.2**
18									1		**9.0**
19											**N/A**
20							1				**7.0**

Table 2.2: Lexicographer file distribution for nouns with up to 20 senses. Number in row *i*, column *j* is the number of words that has *i* senses which fall into *j* lexicographer files.

LFs → senses ↓	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Average num. of LFs**
1	5948												**1.0**
2	1035	1464											**1.6**
3	222	525	338										**2.1**
4	56	204	248	72									**2.6**
5	18	73	144	94	28								**3.1**
6	10	27	66	69	23	3							**3.4**
7	2	14	45	26	24	7							**3.7**
8		3	14	28	29	3	5	1					**4.4**
9		1	6	8	15	7	3	1					**4.8**
10		2	5	9	13	14		1					**4.8**
11		2		4	9	9	3	1					**5.3**
12				4	1	6	3	2	1				**6.1**
13			2	2	4	7	5	2					**5.8**
14				3	1	2							**4.8**
15					5	2	3		2	1			**6.6**
16							3	1	1	1			**8.0**
17						1	1			1			**7.7**
18							1	1					**7.5**
19						1				1			**8.0**
20												1	**12.0**

Table 2.3: Lexicographer file distribution for verbs with up to 20 senses. Description as per table 2.2.

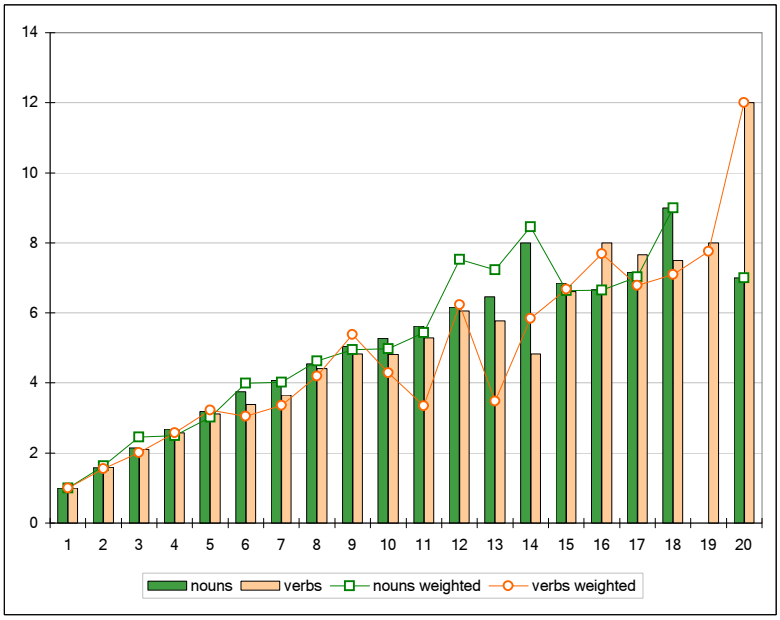

Figure 2.3: Distribution of average number of WORDNET LFs with the number of senses, for the nouns and verbs with up to 20 senses. Bars show the simple average, and the lines show the weighted average according to the corpus frequency of each word in the SEMCOR corpus. The distribution shows that for most words, primes do not include more than two senses; this ensures that too many senses do not get lost by using primes.

Table 2.1 shows the set of Lexicographer files for nouns, verbs, adjectives and adverbs: the latter two parts of speech do not have semantically useful organization for LFs. Tables 2.2 and 2.3 show how the statistics of the number of lexicographer files compared with the number of senses for nouns and verbs. Figure 2.3 show the averaged statistics for the same.

2.3 Semantic Similarity

The notion of semantic similarity existed outside WSD for a long time. Most of the early work, however, were based on word similarity rather than the word sense similarity, ('word sense' as defined in, say, WORDNET, with the word and the sense number). Some early examples for this are those of Miller and Charles (1991), and Rubenstein and Goodenough (1965).

This approach seems to assume somewhat implicitly that when two words are compared, the 'related' senses are automatically selected. This seems to be the case in human thinking as well; For instance, if we ask a person to compare *crane*, *car* and *bulldozer*, (say, as opposed to *crane*, *pelican*, and *eagle*) the machine sense of *crane* will be automatically selected and he would most likely pick *bulldozer* and *crane* as more related to each other, without questioning which sense of *crane* was meant. However in WSD, senses are important, and some later work did focus on sense similarity.

2.3.1 Similarity Measures

Semantic similarity of senses is a widely researched subject. In particular, there are many measures of semantic similarity available for the WORDNET senses. Similarity methods can be defined in terms of features of text in terms of context, corpus frequencies, or proximity within the WORDNET hierarchy.

Surrounding Words

One can use ideas similar to those of Yarowsky (1992), using clues from surrounding words to derive a measure of semantic relatedness. Lesk (1986) originally proposed

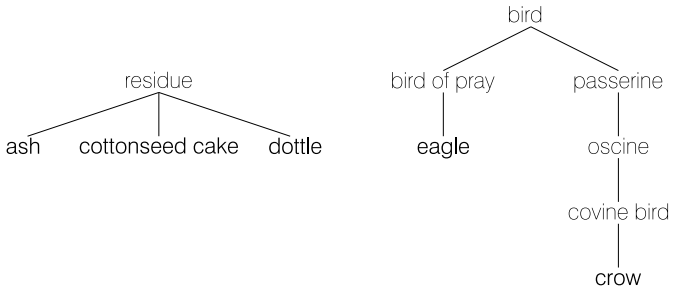

Figure 2.4: Edge distance measure can introduce inaccurate implications. This example shows that *ash*, *cottonseed cake*, and *dottle* being related to each other more than *crow* and *eagle* do.

an algorithm which used electronic dictionary definitions to provide clues. Hence this measure does not depend on the structure of the taxonomy. This algorithm is implemented as a similarity measure in (Banerjee and Pedersen, 2002). This measure seems to be the most vulnerable for noise, as definition texts, short in nature, are not guaranteed to provide solid clues, compared to the hierarchical structure. For instance, the WORDNET definition for *light/1* (visible light) is 'electromagnetic radiation that can produce a visual sensation' and for *elephant/1* (animal) is a mere 'five-toed pachyderm', which hints for a scientific context than the actual contexts the sense is used in day-to-day texts.

Edge Distance

This is the simplest and perhaps the most intuitive measure that uses the hierarchy. In this measure, similarity between two concepts is defined in terms of the path length of traversing from one concept to another in the taxonomical hierarchy (Leacock and Chodorow, 1998). However, this can also lead to wrong implications, as shown in figure 2.4.

Common Parent

The concepts at the lower levels of hierarchy are more specific; if two concepts share a common parent at a lower level of the hierarchy, this means that they are related to each other in more specific details. Hence, the specificity of the common parent can

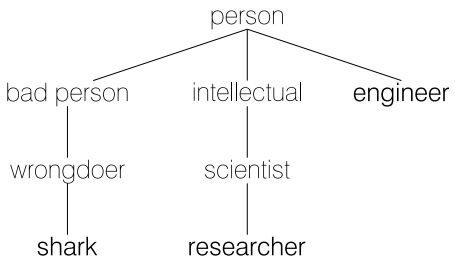

Figure 2.5: Common parent as a measure of relatedness ignores the specific positions of concepts compared. In this example, *researcher*, *engineer*, and *shark* (as a bad person) have similar relatedness to each other.

be thought of as a measure of relatedness. Resnik (1995) used this idea, using the information content of common parent as a measure of similarity between two concepts. Information content is defined in the familiar way, as the negative log likelihood of the concept. Like edge distance, this measure can fail at certain points of the hierarchy, as shown in figure 2.5.

While the Resnik (1995) measure uses only the information content (IC) of the common parent, the Lin (1997) measure uses the formula

$$similarity(c_1, c_2) = \frac{2 \times IC(parent(c_1, c_2))}{IC(c_1) + IC(c_2)}.$$

which accounts for the difference between the specificity of common parent from that of the two concepts compared.

Lexical Chains

The Hirst and St-Onge (1998) measure uses other relations in WORDNET in addition to hypernym IS-A relationships and synonyms. These relations can be either upwards (hypernymy, meronymy), downwards (hyponymy, entailment, etc.) or horizontal (antonymy, attribute, etc.). In addition to these 'strong' direct relationships, 'medium strong' relationships are defined if there is an 'allowable path', a one which links two synsets with a sequence of relations. The notion of what is allowed is a manually made decision. Some of the allowed and disallowed paths are shown in figure 2.6. Arrows towards upwards, downwards and horizontal directions denote the corresponding relations as

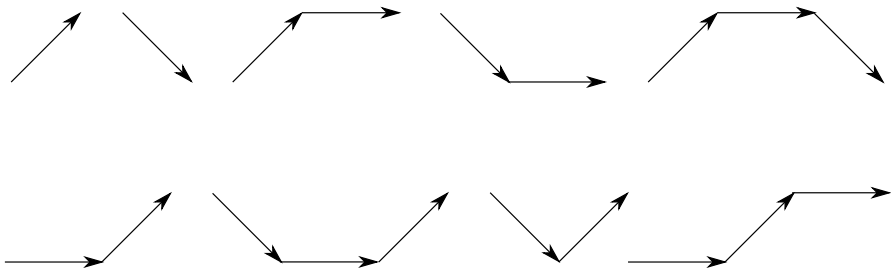

Figure 2.6: Some allowable (above) and not allowable (below) paths for medium strong relations in Hirst and St.Onge measure. Up, down, and horizontal arrows denote up, down, and horizontal strong relationships. See text for details.

mentioned above.

The weight associated to a path is given by

$$weight = C - path\ length - k \times number\ of\ changes\ of\ direction$$

where C and k are constants.

Jiang and Conrath (JCn) Measure

A combination of path length with information content, proposed by Jiang and Conrath (1997), has been shown to perform well in comparative evaluations (Patwardhan, Banerjee, and Pedersen, 2003). Instead of merely taking all children concepts of any given concept as having the same relation with the parent, they introduce a *link strength* of a parent-child link, $LS(c_i|p)$, dependent on the conditional probability of a child concept c_i, given its parent p.

$$LS(c_i, p) = -\log P(c_i|p) = IC(c_i) - IC(p)\ ,$$

where IC is an information content measure defined in the usual way, that is, $IC(S) = -\log P(S)$. This assertion handles straightaway the problem we described about 'odd' children, shown in the *residue-ash* example shown in figure 2.4. Note however, that the 'parent' with respect to a 'child' concept is not defined strictly in the `IS-A` sense of taxonomical hierarchy. JCn can consider other link types, such as meronyms and

holonyms. However the measure as we implemented it uses only hypernym relations.

Let's define $d(p)$ as the depth of node p in the hierarchy, $E(p)$ as the number of edges p has to it's children or *local density*, and $\bar{E}$, the average density of the whole hierarchy. then, the weight of a parent-child edge is defined by

$$wt(c,p) = \left(\beta + (1-\beta)\frac{\bar{E}}{E(p)}\right)\left(\frac{d(p)+1}{d(p)}\right)^{\alpha}[IC(c) - IC(p)]\ T(c,p)\ ,$$

where α and β are parameters such that $\alpha > 0$ and $0 \leqslant \beta \leqslant 1$. These parameters control how much each factor should be weighted. Here, $T(c,p)$ is a measure that determines a similar overall weight depending on the link type. During our experiments we kept α at 0 and β at 1 as the changes required additional validation experiments, with the risk of overfitting, and the difference in performance at the initial cursory tests was not positive.

Once the edge weights are defined, the semantic distance between two word concepts w_i and w_j is calculated as,

$$Dist(w_i, w_j) = \sum_{c \in P} wt(c, parent(c))\ ,$$

where

$$P = \{path(c_i, c_j) - LSuper(c_i, c_j)\}\ .$$

Here, c_i, c_j are the set of possible senses for w_i and w_j, and $LSuper(c_i, c_j)$ is the lowest super-ordinate of c_i and c_j. $path(c_i, c_j)$ denotes the set of nodes between the shortest path from c_i to c_j. Note that JCn measure is originally a distance measure rather than similarity; relatedness between two concepts can be derived from this by subtracting the distance between them by the maximum distance between two nodes in the hierarchy.

Some example relatedness values (calculated using SEMCOR corpus statistics) of *cat*, *horse*, and *shark* to *dog* (all in 'animal' sense) are shown in figure 2.7, along with the relevant tree structure.

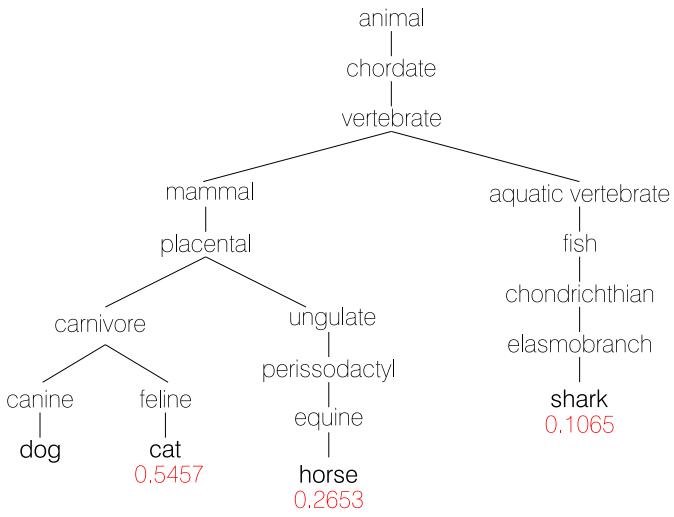

Figure 2.7: A part of WORDNET hierarchy, showing the relative positions of **ANIMAL** sense of words *cat*, *horse*, and *shark* with respect to that of *dog*. The numbers shown are similarity values according to Jiang and Conrath scheme, one model of estimating semantic relatedness between WORDNET concepts.

2.4 A Framework for Class Based WSD

The discussion of the theoretical work and the details of WORDNET structure provided so far is sufficient to outline the basic strategy of the generic sense learning system we proposed in the previous chapter.

The framework we propose for class-based word sense disambiguation is based on a set of assumptions, and is motivated by problems in contemporary unrestricted WSD research, which we discussed in section 1.3. It is meant to test our hypothesis that generalizing word senses across word boundaries is practically possible. For practical reasons regarding data acquisition and evaluation, the experiments use WORDNET senses as the fine grained lexicon. In addition to the practical convenience of availability of training and evaluation data, this is necessary for the sake of argument as well; any formulation that cannot be applied in contemporary WSD problems would not possibly be justified on practical grounds. For this reason, all our experiments are evaluated on fine-grained senses. This requirement can be easily obtained by almost any generic sense class set with only one additional input: a mapping which can convert the classes learned into WORDNET fine-grained senses.

This sense mapping is the central concept in this work. Basically, the mapping is such that each word sense in a given lexicon is assigned a unique coarse grained *sense class*. The class may come with a semantically meaningful label attached, or may just have a nominal label which simply serves as an identifier, without any semantic bearing. More than one sense in a given word, and senses in different words, can map to any given class; the number of classes is much smaller than the number of senses within the lexicon.

We initially assume that these classes can be learnt by a suitable supervised machine learning algorithm, using contextual features of text. This approach is not essentially different from the typical fine-grained WSD setting. Any labeled training example meant for fine grained WSD can be automatically converted to a labeled example for class learning, as we have a direct mapping from senses to classes. However, unlike the case of traditional lexical sample approach to WSD, the fact that senses from different words may map to a single class means that we can use training examples labeled with *different word senses* as training examples for a particular class. This pool of examples, in theory, will be useful as training examples for all words that have one or more sense falling into that particular class.

This system can be used to classify an instance of a word in a given context, into its respective class. However in fine-grained WSD, we need the fine grained sense label. This is not straightforward even though we know the underlying word we just classified, because more than one sense of the word can possibly map to the same class. However, as described in section 1.3.2, word senses have rather skewed distributions; WORDNET senses are ordered in descending order of their frequencies, and successive senses have rapidly falling frequencies. This means that if a given class maps to two senses, the sense with the smaller sense number has a higher likelihood to be the actual underlying sense. This provides a heuristic to create an inverse-map from classes into fine grained senses; pick the sense with the highest likelihood of occurring, using a predefined order. This is the same heuristic we utilized in section 1.3.2, using WORDNET lexicographer files as generic classes. This mapping is obviously lossy as some senses will be discarded.

Admittedly, we depend on the WORDNET sense order for reverse mapping. However, this is not a problem in itself, as supervised systems are allowed to use the sense ordering; This is an informative feature about senses in practice, and there has been several recent works which were focused on the issue of deriving the sense order or the relative sense frequencies for different corpora (Koeling, McCarthy, and Carroll, 2005; Chan and Ng, 2005). Other alternative of picking a random sense given the class would not have made any practical sense, as selecting the maximum likelihood estimate is straightforward even by using our own training dataset, which happens to be SEMCOR, which would theoretically yield the same result as using the WORDNET sense order.

One important note is that we did not commit ourselves to a particular set of classes in the definition. We will use WORDNET lexicographer files for illustrative purposes, but any grouping of senses can serve as the set of classes as long as it satisfies the condition that fine-to-coarse mapping assigns each fine grained sense to only one class.

This system can effectively increase the available amount of training data through example *re-use*. However there are certain complexities that arise in practice. For the sake of clarity we defer discussing these reasons in detail until section 3.2; the relevant fact is that not all different word sense instances can be trusted to provide the clues of same quality for a given word. For instance, the word *amoeba* can not be trusted to provide a good training example for the 'animal' sense of word *dog* as word *horse* could, because of the obvious differences in contexts they appear.

We introduce the use of measures of semantic similarity in order to handle this issue. We show that classifiers can be biased to prioritize different training examples depending on their similarity to the target word being classified.

In summary, the framework for learning sense classes for fine grained WSD consists of:

- a sense-to-class map that is used to transform available fine-grained sense labeled training instances into class-labeled instances
- a classifier system that labels unseen instances into sense classes, using contextual

sense 1 ARTIFACT: a permanent structure that has a roof and walls (48)

sense 2 ACT: the act of constructing or building something (3)

sense 3 ACT: the commercial activity involved in constructing buildings (1)

sense 4 GROUP: the occupants of a building (0)

Figure 2.8: Lexical file mapping for noun *building*. Shown within the brackets are frequencies of each sense in the SEMCOR corpus.

features and with the aid of semantic similarity between instances

- a heuristic to convert the newly labeled instances to fine grained word senses, with some loss in accuracy during the process, due to the many-to-one nature of the sense map.

Chapter 3 will detail the actual implementation of this framework. In the rest of this chapter, we will define some terms that will facilitate discussions throughout the thesis, and discuss some related work done in the area.

2.5 Terminology

In this section, we present a set of terms that we will be using in the following discussions. For easy understanding, recall the example we provided in section 1.3.2. This is shown again in figure 2.8.

2.5.1 Sense Map

The most important idea is the notion of a sense map, which relates the fine-grained senses with coarser classes.

Definition 2.5(a): Sense Map

A sense map $\phi : \mathcal{S} \rightarrow \mathcal{C}$ is a many to one mapping from the set of fine-grained word senses $\mathcal{S}$ of a given lexicon into a set of classes $\mathcal{C}$. For each word w_i in the lexicon, each sense $s_{i,1}, s_{i,2}, s_{i,3}, \ldots, s_{i,n}$ of w_i is a unique element in $\mathcal{S}$. Any given element in $\mathcal{C}$ may be mapped to from senses of different words as well as from several senses of a given word.

In the example in figure 2.8, the underlying sense map is the WORDNET lexicographer file mapping. $\mathcal{S}$ is the set of WORDNET senses and classes in $\mathcal{C}$ are WORDNET lexicographer files.

2.5.2 Sense Ordering, Primary and Secondary Senses

Next we need an ordering which determines which senses we can throw away during the coarse-to-fine sense transformation.

Definition 2.5(b): Sense Ordering

A sense ordering $\prec$ is a partial order on a set of word senses $\mathcal{S}$ where $s_{i,p} \prec s_{i,q}$ means that the estimated frequency of $s_{i,p}$ in a given corpus is higher than or equal to that of $s_{i,q}$. The senses $s_{i,p}$ and $s_{j,q}$ are comparable if and only if $i = j$.

In the example in figure 2.8, the ordering is given by SEMCOR corpus frequencies, and senses 1, 2, 3, and 4 follow this order. In general, in the case of WORDNET senses (and the SEMCOR corpus), the ordering is simply the numerical order of senses. We can see that sense 3 will have to be discarded in favor of sense 2. We will call sense 2 the primary sense in class **ACT** for word *building*.

Definition 2.5(c): Primary Sense

Given a sense a map $\phi : \mathcal{S} \rightarrow \mathcal{C}$, an ordering $\prec$ on $\mathcal{S}$, a word w_i with senses $\mathcal{S}_i = \{s_{i,1}, s_{i,2}, s_{i,3}, \ldots, s_{i,n}\}$, and a class $c \in \phi(\mathcal{S}_i)$, the primary sense of w_i within c, denoted $Prime(w_i, c)$, is $s_{i,j}$ such that $\phi(s_{i,j}) = c$, and $j \prec k \; \forall k \neq j$ such that $\phi(s_{i,k}) = c$. A primary sense of w_i (without respect to a class) is any sense $s_{i,j} = Prime(w_i, c)$ for some class $c \in \phi(\mathcal{S}_i)$.

Definition 2.5(d): Secondary Sense

Any sense that is not a primary sense of w_i is a secondary sense of w_i. The primary sense of a given secondary sense $s_{i,k}$ of w_i is $Prime(w_i, \phi(s_{i,k}))$. A secondary sense of a given primary sense $s_{i,j}$ of w_i is any sense $s_{i,k}$, $k \neq j$, such that $\phi(s_{i,k}) = \phi(s_{i,j})$, or in other words, $s_{i,j} = Prime(w_i, \phi(s_{i,k}))$. A secondary sense of a class c is any secondary sense $s_{i,k}$ of w_i such that $\phi(s_{i,k}) = c$.

It follows that, under this particular sense map, the senses 1, 2 and 4 are primary senses for the word *building*. Sense 3 is a secondary sense, with sense 2 as its primary sense.

2.5.3 Sense Loss

This is a one important factor that determines the practical utility of a given sense map. It denotes how much information is lost by a given sense mapping, in terms of proportion of instances that lost its original senses. Instances lose their original sense during the reverse mapping, if they originally belonged to a secondary sense of the word under a give class map.

Definition 2.5(e): Sense Loss

Sense Loss due to a mapping ϕ in a given labeled corpus is the proportion of corpus instances that are labeled with secondary senses of the respective words, under the mapping ϕ.

In the example, we see that one instance that belongs to sense 3 of *building* loses its original sense, out of total 52 instances, or $\frac{1}{52}$th of the total . The loss for the SEMCOR corpus with respect to the WORDNET LF mapping can be calculated by summing these numbers over all words and taking the result as a fraction of the total number of labeled instances.

Another interpretation of sense loss provides a better practical bearing: suppose we have a classifier that can classify any unseen sense instance (of a given test set) into coarse-grained class level at 100% accuracy. When we use the reverse-mapping heuristic explained above, this classifier can be used as a fine-grained WSD system; this WSD system will make an error for any instance of the test set that is labeled with a secondary sense with respect to the given class map. Sense loss of the class map in the test set is the percentage error the system would make on the test set.

2.6 Related Work

Word sense disambiguation had its own colorful array of approaches. The earliest approaches of the problem date back to more than half a century, with Warren Weaver's famous question on how much context one would need to identify a meaning of a word in a text (Weaver, 1949).

Interestingly, but perhaps not much differently from any area of research with a long-enough history, traces of the techniques that appear in the early work can be seen in several much recent research, at least at the theoretical level. What determines the agenda of a particular time somewhat depends on the general technical focus at that time, as well as new developments in the area —such as the availability of cheap computing power and new lexical resources— rather than new theoretical findings alone.

In this light, it is not easy to present an exhaustive description of approaches used in WSD. We feel a better approach would be to address the history of research that were concerned on the issues that we address in this thesis, after a brief mention on the current state of the art in WSD in general.

Supervised learning clearly dominates the state of the art, offering the best performance. All systems that reported top performance on recent SENSEVAL tasks (Edmonds and Cotton, 2001; Snyder and Palmer, 2004; Mihalcea, Chklovski, and Kilgarriff, 2004) used supervised learning. Some systems showed consistently good results as well; Hoste, Kool, and Daelmans (2001), and Decadt et al. (2004) achieved the second best and top performance respectively on SENSEVAL-2 and 3 English all-words tasks, with the same system (with some modifications in SENSEVAL-3). A similar case in English Lexical sample task is (Lee and Ng, 2002) which reported performance figures that exceeded the official figures reported in SENSEVAL-2 task, and could keep similar performance levels (Lee, Ng, and Chia, 2004) in SENSEVAL-3 as well; it was placed third, with only slightly lower performance than the top system.

These methods, however, do not claim to address the issue of the knowledge acquisition bottleneck. In the case of lexical sample systems, the data provided were used; and GAMBL system mentioned above (Decadt et al., 2004) used in the English all-

words task whatever the training data it could muster from various sources, including the training examples provided for the lexical sample task.

As the condition that motivated our research is the scarcity of training data, we shall focus on approaches that mainly dealt with systems that either do not use training data, use encoded knowledge sources, or employ generalizing schemes on available knowledge. Some of these approaches, generalization schemes in particular, were already discussed in detail in chapters 1 and the early sections of this chapter, so our handling of them here will be limited for just references. Interested reader is advised to refer to section 2.1 for more information on generalization schemes.

2.6.1 Some Early Approaches

Sense Knowledge can be encoded in different kinds of structures, as the knowledge of word senses is essentially a formalization of how humans see the world. One can suggest that a system which contains all the underlying rules can be used to understand language and then disambiguate senses. Another more shallow approach is to resort to a dictionary lookup. The latter approach does not require *training data* per se for statistical machine learning, and in this regard, can be thought of as a knowledge generalizing approach. This section will present a few miscellaneous and diverse set of approaches that are not commonplace in current state of the art, before moving into more recent and related work from section 2.6.2.

Symbolic Knowledge Encoding

Indeed, one does not deny the requirement for world-knowledge based inference (as opposed to lexical and syntactic features of text) in at least some cases of sense disambiguation. One good example for this was put forward by Bar-Hillel (1964); this was discussed earlier in section 1.1.2.

Symbolic AI methods try to tackle this using networks of concepts which can inherit behavior and properties from more primitive 'super' concepts. Early works of this kind includes (Masterman, 1961) and (Quillian, 1969). Another approach that tackled

WSD problem through symbolic AI line is the *preference semantics* (Wilks, 1968; Wilks, 1975). This is an approach of logic for language understanding, which uses semantic primitives (Schank, 1973) for reasoning about word senses. This is a scheme of generalization in a way, as the primitives are essentially basic building-blocks of knowledge who can inherit properties from their semantic parents. Wilks (1975) also discusses about the problems with a rigid human-coded class system: an inflexible set of primitives is of less use in terms of versatility, and so-called primitives are not essentially distinguishable from non-primitive word meanings, as any concept has the potential to be a primitive to more specific concepts, while having more generic concepts as its own primitives.

Encoded Knowledge: Machine Readable Lexico-Semantic Resources

Compared to the techniques discussed above, somewhat more current, popular, and direct in WSD research are the forms of knowledge encoding in dictionaries, glosses, and other forms of databases.

The basic idea behind the dictionary based systems is quite simple, and analogous to what a human reader would do when looking up the dictionary to find the meaning for an unknown sense. However, unlike the symbolic AI based systems that use techniques such as case frames and reasoning, the surface level similarities between the context of the target word and the dictionary definition, or *gloss* are considered.

The WSD algorithm proposed by Lesk (1986) is one early and notable attempt in this regard. It compares the *gloss* (dictionary definition) of each sense of the word being disambiguated with the glosses of other words in the context. The sense whose gloss yields the highest match is selected as the correct sense of the word. While this technique in itself does not generalize any knowledge, it has been extended by Banerjee and Pedersen (2002) to utilize the hierarchical relatives from each synset; the glosses of the relatives allow more information than the gloss of the sense itself provides. This latter approach has some generalizing nature in it, as it borrows knowledge from the hierarchical neighbors.

Another major, perhaps the largest ever, approach based on systematic human en-

coding of knowledge is CYC (Lenat, 1995), which claims to build a knowledge structure that would complement an encyclopedia by providing the necessary semantic annotations for reasoning. The system is commercial, and a separate but related project OPENCYC is an adaptation that is non-commercially available in GNU Lesser General Public Licence. CYC contains limited support for WORDNET senses. At least one research work (Curtis, Baxter, and Cabral, 2006) reports on the application of CYC for word sense disambiguation.

WORDNET itself is this kind of a database. An interesting comparison and critique between similarities and differences of the approaches of CYC and WORDNET, written by their respective creators, is (Lenat, Miller, and Yokoi, 1995).

SENSEVAL attempts that used dictionary-based approaches include (Litkowski, 2001; Litkowski, 2000).

2.6.2 Generic Word / Word Sense Classes

As we described in earlier in this chapter, the approaches for overcoming data sparseness by generalizing properties of word senses can be broadly divided into two types, class based and similarity based, depending on whether they assume a class-structure.

Class Based Models

Class based models consider words that show similar behaviors as belonging to the same *class*. They can either work on a set of classes with already identified semantics, or derive a set of classes on their own. One earlier example for this kind in word sense disambiguation is (Yarowsky, 1992), which categorized words according to the global context they appear in. Yarowsky used Roget's thesaurus categories, which were about 1000 in number, as his set of classes, and used Groiler's Encyclopedia in order to learn the 'salient features' of classes. Another example is Selectional constraints (Katz and Fodor, 1963), which were adapted for WSD by Resnik (1996; 1993). Both of these systems depend on a manually created set of classes. (McCarthy and Carroll, 2003) is another similar application. (Ciaramita and Johnson, 2000) also used selectional pref-

erences, but with Bayesian networks, which is not very common in the WSD literature.

Brown et al. (1992), on the other hand, try to infer the set of classes from the features found within the text. They argued that the words that have similar usage could be put into classes, even when we don't have a classification defined beforehand. For instance, the words *Thursday* and *Friday* can be expected to have same probability patterns with other words, although there can be exceptions such as that *'thank God it's Thursday'* does not appear commonly. They used n-grams as a learning model.

In a way, their work is much similar to ours on the argument level, as they also claim that similar words have similar contexts. On the other hand, this observation is a generic one, and crucial differences lie at the application level. Brown and colleagues, for instance, work on similar *words* rather than similar *senses*. Other researches that were based on similar approaches are (Pereira, Tishby, and Lee, 1993; Pereira and Tishby, 1992). Magnini and Cavaglià (2000) also introduce a generalization much similar to Yarowsky's approach, but is defined on WORDNET senses. The topic signatures they introduce are called subject field codes, and are based on the contexts (*domains* in their words; see also (Gliozzo et al., 2004)) they appear in, rather than a common taxonomical class they belong to.

Similarity Based Models

Some other works tried to utilize information available from the words that are contextually similar, but without assigning them into any defined class. One representative system of this kind is done by Dagan and colleagues (Dagan, Marcus, and Markovitch, 1993; Dagan, Pereira, and Lee, 1994). They argued directly against class-based approaches (Dagan, Lee, and Pereira, 1997) on the grounds that it can cause the idiosyncrasies of the individual words to be ignored. As an example they took color words: for instance, although *red* can act as a generic **COLOR** word, it has very distinctive co occurrence patterns that cannot be found in other words, such as the relation with words such as *apple*.

Work from Classic Linguists

The same arguments has been considered outside Computational Linguistics. One of the most famous of verb classification systems is the set of classes proposed by Levin (1993) for English verbs. They are based on commonalities in diathesis alterations, or the alterations between the expressions and arguments the verbs participate in. As an example, consider the sentences

a. Mary cut the bread.

b. Jane broke the glass.

c. Tom touched the cat.

and

a. Bread cuts easily.

b. Glass breaks easily.

c. * Cats touch easily.

The fact that *touch* does not allow the alteration from the first form to second form is a determining factor in deciding that *touch* does not belong to the same class *cut* and *break* do. Although these classes are based on syntactic properties unlike those in WORDNET, it has been shown that they can be used in automatic classifications (Stevenson and Merlo, 2000). Korhonen (2002) proposed a method for mapping WORDNET entries into Levin classes. Olsen, Dorr, and Clark (1997) also describe an attempt to link WORDNET synsets with Levin's classes. To the best of our knowledge, no research has been done on Levin classes' applicability in WSD.

Coverage-wise, Levin's classes are fairly large, but not as comprehensive as WORDNET . There are 193 verb classes in total, which cover 3100 verbs. The classes, like the experiments of Miller and Charles (1991) and Rubenstein and Goodenough (1965) we mentioned earlier, are based on *verbs* rather than *verb senses*; there is no distinction among different senses of the same word, so the utility in WSD is somewhat limited.

Wierzbicka (1996) also argued that the taxonomical classification of concepts do not necessarily have to conform to their linguistic usage (Wierzbicka, 1984), and that the usage based 'classes' can be different from a taxonomy. Her examples provided some of the inspiration for our work, although her work, as Levins', were strictly of classic linguistic nature, and were not concerned about any computational approaches to the problems. Additionally, both of them focused on *words* rather than word senses, as was the case with Brown mentioned above, and those of Miller and Charles (1991) and Rubenstein and Goodenough (1965) we mentioned in section 2.3.

Learning WordNet Lexicographer Files

Learning the WordNet top level concepts as semantic classes has been attempted by several researchers for different uses. Ciaramita and Johnson (2003) used a multi-class perceptron tagger for classifying noun instances into WordNet lexicographer files. They assumed that the key differences in semantics are held at super-sense level; their approach used WordNet hierarchy for creating annotated instances for training. Curran (2005) implemented a similar system, but using unsupervised learning, and used a vector-space similarity based approach. Both of these systems were limited to WordNet nouns only, and were not concerned with word sense disambiguation.

In the context of WSD, Crestan and colleagues (Crestan, El-Bze, and Loupy, 2001; Crestan, 2004) employed an approach similar to ours in Senseval English lexical sample and all words tasks, by learning WordNet lexicographer files as classifier level and then converting them into fine grained WordNet senses. However, they do not try to utilize the fact that the classes at classifier level are generic for several senses, hence do not claim to use training examples from different words to learn a class. They do not exploit the notion of semantic similarity to obtain better substitute training examples either. The idea of using WordNet LFs is still useful, as it reduces the granularity of the senses, making them easy to learn from a limited amount of training data.

2.6.3 Clustering Word Senses

Most research described in the previous section utilized an already present set of classes. In our work, we tried to identify clusters of senses from within the contexts. In outline, this idea has some relationship to the strong contextual hypothesis of Miller and Charles (1991): that the semantic similarity between a pair of words can be determined by the extent of similarities in the contexts they appear. The use of clustering techniques was the typical approach used in many unsupervised work, which discriminated word senses into clusters depending on the context of each instance. (Pedersen and Kulkarni, 2005) is a representative system which employs this approach; given a set of word sense instances, the system creates vector-based representations of contextual features and performs clustering on this instance base. The authors claim that the framework can be used for other purposes than WSD.

Agirre and de Lacalle (2003) used on SENSEVAL-2 lexical sample data, several clustering schemes that use information other than context feature vectors, such as similarity matrix of word senses —based on classifier confusion matrices— to cluster fine grained WORDNET senses into coarse grained classes. The results of the clustering is evaluated using the coarse grained senses provided in SENSEVAL task, and was shown somewhat promising: more than 80% 'purity', measured by which proportions of different instances from manually-categorized coarse grained senses are included in a given cluster generated by the algorithm.

We use Singular Value Decomposition for a clustering system we introduce in chapter 6. Although not quite similar to our approach, Strapparava, Gliozzo, and Giuliano (2004) used the SVD-based technique Latent Semantic Indexing in identifying the semantic domains. The relevant technique for WSD is presented in (Gliozzo et al., 2004).

2.6.4 Using Substitute Training Examples

Most works that employed substitute training examples in order to alleviate the knowledge acquisition bottleneck, utilized information from the WORDNET hierarchical neighbors of the senses of the target word. Some approaches used information derived from

the neighbor instances to augment the features or clues for disambiguating a given sense, while some others directly used instances of the neighbors as makeshift examples for training.

The first approach is employed by Mihalcea and Moldovan (2000) in an iterative algorithm for word sense disambiguation. The method involves an algorithm which starts with a few instances that can be disambiguated with high confidence, and then uses a gradually relaxing linking process of unlabeled examples with already labeled examples. The linking is done using WORDNET hierarchy relations. For instance, if there is an instance of *authorize* that has been disambiguated as sense 1, and an unlabeled instance of *clear* in its proximity, the latter is marked as *clear/4* considering the fact that *authorize/1* and *clear/4* are in the same synset. Each iteration uses similar kind of a heuristic that is weaker than the previous in terms of relational strength, hence in confidence.

Some work of Leacock and colleagues (Leacock, Miller, and Chodorow, 1998; Leacock and Chodorow, 1998) provide examples for the latter approach of automatically gathering training examples using WORDNET relationships. In (Leacock and Chodorow, 1998) they used proximity based hypernym relationships for learning useful contextual patterns that were not found with training examples for the original word. In (Leacock, Miller, and Chodorow, 1998), monosemous relatives of word senses are extracted from an unlabeled corpus, and are used as training examples for respective senses of polysemous words.

Other examples for the latter approach include (Agirre and Martinez, 2004; Agirre and Lopez de Lacalle Lekuona, 2004).

2.6.5 Semantic Similarity

The use of WORDNET neighbors to help address the knowledge acquisition bottleneck, in itself without any explicit use of similarity *measures*, entails the notion of semantic similarity. This is because the neighbors of a concept in WORDNET hierarchy are, in general, semantically close to the concept of the primary interest. However, the explicit use of the notion of semantic similarity was not seen in WSD research until recently.

Some early studies of semantic similarity include the work of Miller and Charles (1991). They claimed that "the more often two words can be substituted, the more similar in meaning they are judged to be". This idea is somewhat similar to what we exploit when we use substitute sense examples from different words in classifier process, described in section 3.2. Miller and Charles used noun pairs (as opposed to demarcated noun *sense* pairs) in their experiments. Another widely-cited work along this line is (Rubenstein and Goodenough, 1965). The authors claimed that common contexts between two words is indicative of their similarity in meaning.

The idea is extensively used by Ted Pedersen and colleagues (Patwardhan, Banerjee, and Pedersen, 2003; Banerjee and Pedersen, 2002) whose work also paved the way for popular open-source implementation of many semantic similarity measures in Perl programming language, available through the GNU general public licensed software library `WordNet::Similarity` (Pedersen, Patwardhan, and Michelizzi, 2004).

Their disambiguation algorithm is adapted from the Lesk (1986) algorithm mentioned above. The candidate senses of the word being tested are compared for similarity with those of the words in its context. The sense that gives the best total similarity score is picked as the correct sense. Their extension allows the algorithm to be used flexibly with other similarity measures as well as the modified version of the original Lesk (1986) measure of gloss overlap; they reported best performance with the adapted Lesk measure.

Note that their method is radically different from ours in the way that it uses the similarity measure itself as the disambiguation rule, while in our case, similarity measure is merely an indicator of the *reliability* of a particular training instance.

2.7 Summary

Sense generalizing schemes have been a recurring subject in NLP under various contexts. This section provided an introduction to sense generalizing schemes – systems that can be used to infer knowledge about different words, with senses that show similar behavior. In section 2.2, we discussed the basic structure of WORDNET , and showed

how it provides a basis for sense generalizing. Section 2.4 outlined the necessary features of a framework for learning generic sense classes, with fine-grained WSD as the end objective. Section 2.5 was devoted to define a set of terms dealing with the properties of this framework. Several related research work were discussed in section 2.6.

In the next chapter, we will discuss a system which implements the framework we discussed in section 2.4, with WORDNET lexicographer files as generic sense classes.

It is the duty of every citizen according to his best capacities
to give validity to his convictions in political affairs.
— Albert Einstein
Treasury for the Free World 1946

Chapter 3

WordNet Lexicographer Files as Generic Sense Classes

With the basic framework for generalizing word senses into coarser level classes, one obvious fact is that the set of WordNet lexicographer files makes an intuitive candidate for generic coarse grained classes. In the previous chapter, we introduced the conceptual framework of a sense class learner, without discussing any technical details. When it comes to implementation, a great deal of ideas can be borrowed from typical WSD systems; however, there are many possible ways WSD systems are implemented, so the design is not a trivial problem.

We describe in this chapter our implementation of the sense-class learning WSD framework, justifying the reasons for our technical decisions as and when necessary. As mentioned earlier, we tackle the problem in two steps: first we try to disambiguate word instances into sense classes at WordNet lexicographer file level, using labeled training examples and a supervised learning algorithm. Then we transform these classes into fine grained senses.

Parts of this problem were the focus of several previous research work; for instance, Ciaramita and Johnson (2003) and Curran (2005) addressed the first part of the problem, by attempting to classify nouns into WordNet lexicographer files. However these experiments were limited to nouns, and they did not evaluate any utility of these

classes at the application level. Crestan (2004) used a system that learned word sense at WORDNET LF level in SENSEVAL-3, but made use of only the coarse-grained level, not the generic nature of classes. Crestan (2004) and Crestan, El-Bze, and Loupy (2001) used WORDNET lexicographer files in fine grained WSD. However theirs was an attempt to gain from the resulting coarse-grain classes rather than to generalize knowledge across different word senses. They do not employ the use of semantic similarity measures to validate training examples from different words, which is a salient feature of our implementation, which shall be discussed later in section 3.2.

The main purpose of this chapter is to describe the basic technical details of our implementation of the coarse-grained sense class classifier framework, which we described in section 2.4. The framework can use any mapping of fine grained senses into generic coarse gained classes; we are going to demonstrate the use of system by using WORDNET lexicographer files as sense classes. Other than the case-specific details pertaining to the demonstration, the framework applies to experiments discussed in future chapters as well, in particular chapter 6. In this demonstration implementation, our experiment included both nouns and verbs. As the adjectives and adverbs do not have useful WORDNET LF labels,[1] the LFs cannot be used for either adjectives or adverbs.

3.1 System Description

Our system consists of three independent classifiers that work on three different types of features within text. By design, it is not much different from contemporary systems that use memory based learning.

3.1.1 Data

For training, we used the SEMCOR corpus (Landes, Leacock, and Tengi, 1998). SEMCOR is a manually labeled corpus of 352 files from the *Brown Corpus of Standard American English*, commonly known as the Brown corpus (Francis and Kucera, 1982). It contains

[1] adjectives have only three classes, which are not based on semantics, and all adverbs fall into a single class.

Category	Description	Brown	SEMCOR
A	Press: Reportage	44	44
B	Press: Editorial	27	27
C	Press: Reviews	17	17
D	Religion	17	17
E	Skill And Hobbies	36	31
F	Popular Lore	48	28
G	Belles-Lettres	75	28
H	Miscellaneous: Government & House Organs	30	20
J	Learned	80	49
K	Fiction: General	29	29
L	Fiction: Mystery	24	18
M	Fiction: Science	6	6
N	Fiction: Adventure	29	17
P	Fiction: Romance	29	12
R	Humor	9	9
	Total	**500**	**352**

Table 3.1: SEMCOR corpus statistics: SEMCOR contains disproportionate numbers of documents from the 15 different text categories. Numbers shown under Brown and SEMCOR are numbers of files for each category in the corpus.

one-million word text from sources printed in 1961; the sources are diverse, and the proportions of each are shown in table 3.1, together with which parts of them made their way into SEMCOR. In SEMCOR, the Brown-1 and Brown-2 parts have all open-class words manually labeled with WORDNET senses, while the Brown-v part has only verbs labeled.[2] The complete corpus has 352 files in total, divided into three parts as follows:

part	contents	what's tagged
brown1	103 Brown Corpus files	All open class words
brown2	83 Brown Corpus files	All open class words
brownv	166 Brown Corpus files	Verbs

For evaluation, English All Words task test data in SENSEVAL-2 and 3 exercises were used.

A randomly picked portion (5000 instances per each part of speech) was set aside from training data as generic validation data. For word level validation purposes (used in weighted majority voting, described in section 3.3.1), we set aside a randomly picked

[2]We used a version that use WORDNET 1.7.1 senses, created by Rada Mihalcea at University of North Texas, by automatically mapping from the original version which used WORDNET 1.6 senses.

sample, up to a maximum of 20 instances for each word, from training data. In order to avoid large number of instances being removed this way, the development sample for one word is used as training data for other related words. The complete SEMCOR corpus has 82616 labeled noun instances and 92875 verb instances. The greater number of verbs is due to the contribution of `brown-v` part of the corpus to verb examples.

Since our proposition is that the generic classes can be learnable using training examples from different words, we do not restrict ourselves to labeled data of the same word, but use any labeled word instance, if the labeled sense belongs to an LF which includes some sense of the target word being classified. For instance, both *horse* and *dog* share the LF **ANIMAL**, so an instance labeled with **ANIMAL** sense of *horse* can be used as a training example for *dog*.

3.1.2 Baseline Performance

The accepted baseline for supervised systems in WSD tasks that use WORDNET senses is the accuracy of a system that always predicts the WORDNET first sense of a word.

In the English all words task, this is a fairly high level of performance, usually above 60% on average. This is because the sense distribution in English is skewed and a few senses that are used often occupy a disproportionate amount of word instances.

3.1.3 Features

There is a fair agreement on what sort of features are generally useful in WSD. These include topical context, collocations, parts of speech and various syntactic clues, such as noun-verb relations. In our experiment we used the local context, parts of speech and syntactic pattern features.

Local Context

Local context information has been used in different styles in WSD literature. Lee and Ng (2002), for instance, used explicit collocations by using strings of combined words at different relative positions with respect to the test word. Hoste, Kool, and Daelmans

An/DT ancient/JJ stone/NN church/NN stands/VBZ
amid/IN the/DT fields/NNS ,/, the/DT sound/NN of/IN bells/NNS
cascading/VBG from/IN its/PRP$ tower/NN ,/,
calling/VBG the/DT faithful/NN to/TO evensong/NN ./.

Figure 3.1: A sample sentence with parts of speech markup

(2001) used a window of words and parts of speech surrounding the tested word. We used the latter representation, by using a window of words to both sides of the target word. This way, the 'collocations' can be detected by the classifier implicitly; to avoid confusion with the way other people (Lee and Ng, 2002; Yarowsky, 1993) have used the term 'collocations', we will continue to refer to this feature as local context. The word window was n words to both sides, where $n \in \{1,2,3\}$ was selected by cross-validation. This window size can be seen as rather small; more about this decision is discussed later in this section under 'Topical Context', and in section 4.6.

All words were converted to lowercase, and punctuation was omitted while constructing the vector. An example sentence with parts of speech markup is shown in figure 3.1. For example, the local context vector for the word *church* in this sentence is `[an ancient stone stands amid the]`.

Sentence boundaries were considered when constructing the vector, and in case the window exceeded the boundary, the positions that fall outside the sentence were filled with `NULL` values.

Parts of Speech

SEMCOR data files come with part of speech labels assigned to all words, and these were used as-is. For SENSEVAL data, associated Penn Treebank (Marcus et al., 1994) parse results were provided. These are supplied with the DData sets. POS tags, which are at the leaf nodes of the parse trees, were aligned with the XML data files.

The exceptions on the punctuation and sentence boundaries are similar to those used for local context. For instance, the word *faithful* in the sentence in figure 3.1 has the feature vector `[NN VBG DT TO NN NULL]`.

	Feature	Example	Value
noun	Subject - verb	*an ancient **church** stands*	`stands`
	Verb - object	*the bell calls the **faithful***	`calls`
	Adjectival modifiers	*an ancient **church***	`ancient`
	Prepositional connectors	*the sound of **bells***	`sound of`
	Post-nominal modifiers	*the **sound** of bells*	`of bells`
verb	Subject - verb	*an ancient church **stands***	`church`
	Verb - object	*the bell **calls** the faithful*	`faithful`
	Adverbial modifier	*the bell **rings** loudly*	`loudly`
	Subject - Infinitive - verb	*the bell will **ring***	`bell`
adjective	Normal adjective links	*the **ancient** church*	`church`
	Predicative adjective	*the church is **old***	`church`
	Definitive determiner	*the church is the **oldest***	`church`
	Adverb links	*the church is very **old***	`very`

Table 3.2: Grammatical relations used as features. The target word is shown in ***bold***. Adjective patterns are also provided for future reference.

Syntactic Patterns

These are features that capture more direct patterns among words, such as subject-verb and noun-adjective relationships.

Table 3.2 show the complete list of syntactic pattern features, and examples for each of them.

Since there can be more than one pattern present for a given word instance, a binary vector was used to encode all patterns. Each pattern denotes a given bit in the vector, and only the patterns present in the test data were used in the feature vector in order to minimize its length, owing to computational reasons. This means that we have to create the feature vectors from test and training data files at the same time — a trade off for faster processing time. However this does not impose any significant time delay, as the classifier we use —memory based learning– does not have a lengthily 'training' phase; we the bottleneck in classification is always at the classification phase, and there is not much to gain from off-line training.

For extracting the relationships, all texts were parsed with the Link Grammar parser due to Sleator and Temperley (1991).

Topical Context

One reason we did not use the topical context as a feature is that we observed it can result in conflicting information in generalizations. For instance, consider the following senses of *bank*:

Sense 1 a financial institution that accepts deposits and channels the money into lending activities; "he cashed a check at the bank"; "that bank holds the mortgage on my home"

Sense 2 sloping land (especially the slope beside a body of water); "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"

Sense 4 a building in which commercial banking is transacted; "the bank is on the corner of Nassau and Witherspoon"

Clearly, in terms of the topical context, sense 1 and 4 go together, as both involve the same 'financial' sense of bank. On the other hand, if we think in terms of local context and syntactic patterns, we can expect that sense 2 and 4 will go together, as they both refer to locations (as in the case of "He walked towards the bank" or "The vehicle was stopped at the bank"). In short, most selectional preference style predicates that take a physical location as an argument will treat those two senses similarly. In case of fine-grained sense classifier, this is not a problem as all senses are individually considered, paving way for consistency among features within a sense. But in a class based learning system, local context and topical context could yield conflicting information for generalizing. Linking different concepts according to topical context has left open questions in WORDNET design as well (Fellbaum, 1998a, p. 10).

In addition, topical context feature vectors are typically of very large dimension compared to the local context; although this should not be a big problem for classifiers intended for single-word sense learning —as they can concentrate on context words that occur most frequently with that particular word— the dimension can be prohibitive for a classifier that tries to incorporate information from a large set of different words. Our local context feature vector itself was about 42,000 in dimension; a

wider context would have yielded feature vectors that are several orders larger.

Several research work had also discussed this issue (Lesk, 1986; Martinez and Agirre, 2000; Choueka and Lusignan, 1985; Leacock, Miller, and Chodorow, 1998) which generally suggested that local context information is superior, and a majority of words can be disambiguated using local context alone. Lin (1997) also pointed out that psychological evidence suggests that humans can disambiguate senses given a narrow window of context. For these reasons, it is safe to assume that the missing information in our features is not too significant.

3.1.4 The k-Nearest Neighbor Classifier

The classifier used in central part of the experiments[3] is TIMBL (Daelemans et al., 2004), a system based on a memory-based learner (Daelemans, 1999), also known as example-based, similarity-based, case-based, analogical, or more generically, instance-based learning (Màrquez, 2000). It has been argued that Instance Based Learning has the right bias for most natural language tasks (Daelemans, van den Bosch, and Zavrel, 1999). Several systems that produced good results in past SENSEVAL exercises (Hoste, Kool, and Daelmans, 2001; Decadt et al., 2004; Mihalcea and Faruque, 2004) and other data sets such as Ng and Lee (1996) also used memory based learning.

The classifier can directly use the vector format we described for feature encoding. The learner is founded upon the hypothesis that cognitive reasoning is based on reasoning using the *similarity* of a new observation to stored (in memory) representations of earlier observations, rather than on a set of mental *rules* that were derived from the previous observations. The implementation of memory based learning is typically done as a k-nearest neighbor (k-NN) learner (Cover and Hart, 1967).

The classifier (figure 3.2) does not essentially build a model for reasoning upon reading labeled training example, hence the name *lazy learning*. In the 'training' phase, it merely stores all data instances in memory. In practice, this can possibly include an optional indexing structure for faster lookup during classification. At the classi-

[3]We experimented with support vector machines in order to verify whether it is possible to extend the ideas for other classifiers; this is discussed in section 3.4.

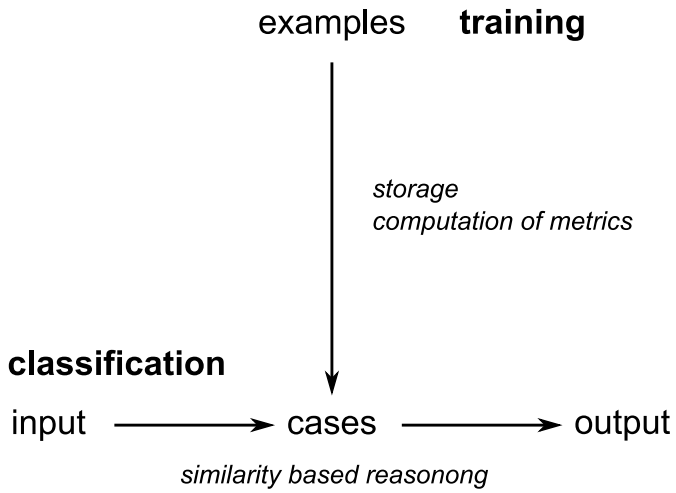

Figure 3.2: Memory based learning architecture (Daelemans et al., 2004)

fying phase, each stored instance is compared with the target instance. For this, the classifier uses a measure of *distance*. Let us assume that the each instance is an n dimensional vector of features. We calculate the distance $\Delta(X, Y)$ between two instances $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ as

$$\Delta(X, Y) = \sum_{i=1}^{n} \delta(x_i, y_i)$$

Where $\delta(x, y)$ is a distance measure defined for values of a given feature. If x, y are numerical features, then the difference (possibly normalized) between the two values is considered as the distance; For nominal feature values, the distance can be either something as simple as 0 for equal values and 1 for different values, or some complex function based on information available from the class distribution.

We initially experimented with several distance measures, using the development dataset for validation. What proved to perform best is Jeffrey divergence: this is a symmetric form of Kullback-Liebler information theoretic measure of 'distance' between two probability distributions (Kullback and Leibler, 1951). Jeffrey divergence is defined for two feature values v_1 and v_2 on a class distribution $\mathcal{C}$ as

$$\begin{aligned}\delta(v_1, v_2) &= D_{KL}(P(c|v_1)\|m) + D_{KL}(P(c|v_2)\|m) \\ &= \sum_{c_i \in \mathcal{C}} \left\{ P(c_i|v_1) \log \frac{P(c_i|v_1)}{m} + P(c_i|v_2) \log \frac{P(c_i|v_2)}{m} \right\}\end{aligned}$$

where

$$m = \frac{P(c_i|v_1) + P(c_i|v_2)}{2}.$$

This measure is claimed to be generally more robust in cases of sparse distributions (Daelemans et al., 2004).

The actual classification is done by inspecting what stored instances are most similar to the instance being classified; as the name implies, k most nearest instances are picked up, and they vote for the final class. Voting power of an instance may be dependent on the distance between itself and the target instance. The class that gets the highest number of votes is selected.

The classifier supports an extensive collection of options, however we did not want to optimize all of them as there is a very high risk of overfitting with the small amount of training data available. Only the basic set of options:

- distance metric to use: among simple overlap, information gain, gain ratio, modified value difference and Jeffrey divergence (see (Daelemans et al., 2004) for information on the measures).
- number of k in the classifier, i.e. the number of nearest neighbors that participate in voting for class label, $k = \{1, 2, 3, 4, 5, 6, 7\}$.
- weight threshold to select example instances (see section 3.2 below)
- the size of context window, $\{1, 2, 3\}$ for local context and part of speech feature vectors.

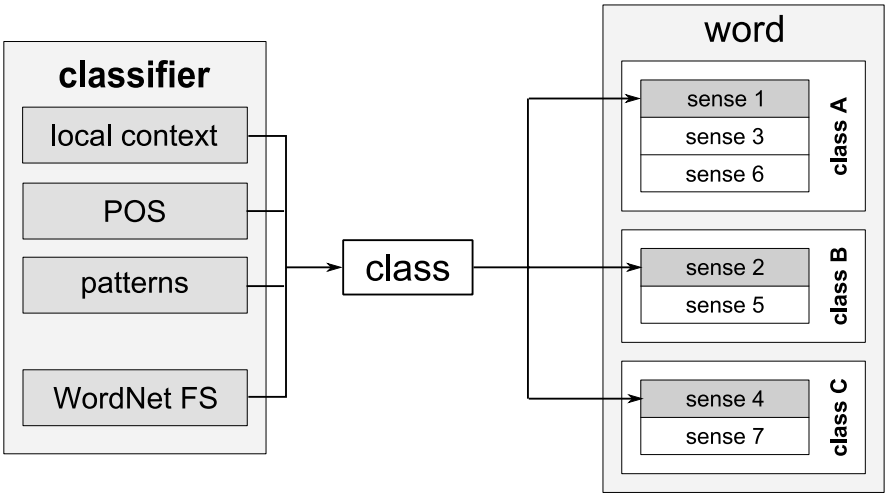

Figure 3.3: Classifier combination and fine-grained sense labeling. The three classifiers and WORDNET first sense participate in voting, which returns a the predicted sense class. This is converted to WORDNET fine-grained sense by picking the relevant primary sense, i.e. the sense with smallest number that falls in the predicted class.

3.1.5 Combining Classifiers

Having individual classifiers that work on different types of features, and combining them through voting, has been shown to perform well in WSD previously (Decadt et al., 2004; Villarejo et al., 2004; Mihalcea and Faruque, 2004). The system we implemented uses this strategy.

The three sets of features (local context, parts of speech, syntactic patterns) are used by three different classifiers, which independently output the class for a given word instance. In addition, a classifier that always predicts the class of the first WORDNET sense of the word is used in voting. Simple majority voting and two weighted majority voting schemes were tested: these are described in section 3.3.

The output from the voting is a WORDNET LF; it is straightforward to convert this to the finer-grained sense; as explained in section 2.4, we do this by selecting the primary sense for each LF as the corresponding fine-grained sense. Figure 3.3 shows a schematic diagram of the system.

3.2 Example Weighting

In the initial experiments it was shown that using all examples for a particular class does not help improve the classifier performance. This is not surprising: As WORDNET LFs are rather coarse, one LF can include many senses that belong to much diverse concepts. For instance, LF **ANIMAL** includes nouns ranging from *amoeba* to *elephant*, including birds and fish. It is clear that an instance of *amoeba* does not help as an example for learning **ANIMAL** sense of *dog* as much as *cat* would do.

For this reason, some constraining is necessary to avoid feature confusion. The necessity for this may better be explained by an example: Consider the first two senses of word *school*,

sense 1 **GROUP**: an educational institution

sense 2 **ARTIFACT**: a building where young people receive education

and two training examples *company* as a **GROUP** *(an institution created to conduct business)*, and *tape* as an **ARTIFACT** *(a recording made on magnetic tape)*. Assume that we have an unlabeled instance of *school* in the context *'run the school'*. Two labeled examples *'run the tape'* and *'run the company'* will provide contradictory clues that would signal for two different sense classes of *school*.

If, however, we consider the fact that **GROUP** senses of *school* and *company* are much similar to each other compared to **ARTIFACT** senses of *school* and *tape*, we can assume *company* example to be more reliable one among two contradictory substitute examples.

This application of similarity weighting benefits the generalization in other ways as well. For instance in a setting where selectional preferences provide the main information (Resnik, 1996), the predicate `object-of-`*`run`* could possibly be ignored totally, as its selectional preference strength is low due to the fact that it results in conflicting signals for classes. However in this implementation, it is possible to use information relevant to both classes by selectively constraining the system to accept only what is meaningful for the target word. This allows the system to use weaker clues effectively.

3.2.1 Implementation with k-NN Classifier

The obvious way to handle this kind of constraining is to introduce a localizing scheme that will bring in some sort of preference, which would say that instances that are more 'similar' to the target word must be treated as more authoritative. In the k-nn classifier we use, this can be accomplished by modifying the distances of the original instance base. More authoritative training examples are moved towards the test instance, and semantically distant examples are moved further away from the test instance.

Assume that the original distance between a training instance X and testing instance Y is $\Delta(X, Y)$. Also assume that the similarity between X and Y is $S_{X,Y}$. Then the distance can be adjusted such that the new distance $\Delta^E(X, Y)$ is given by

$$\Delta^E(X, Y) = \frac{\Delta(X, Y)}{S_{X,Y} + \epsilon}$$

where ϵ is a small constant added to avoid division by zero.

In the practical implementation, the exemplar weights were derived from the following method:

1. pick a labeled example e, and extract its sense s_e and sense class (WORDNET LF in this particular case) c_e.

2. if the class c_e is a candidate class for the current test word w, i.e. the candidate word has any senses that fall into c_e, find out the primary sense of w, s_w^{ce}, within c_e. (Recall that the WORDNET sense ordering is in use; thus, the primary sense is the sense that has the lowest WORDNET sense number within that class.) If none of w's senses fall into c_e, ignore that example.

3. calculate the relatedness measure between s_e and s_w^{ce}, using whatever the similarity metric being considered. This is the exemplar weight for example e.

An example Consider creating training data for the word *'dog'*: this word falls into three sense classes, namely **NOUN.ANIMAL**, **NOUN.PERSON**, and **NOUN.ARTEFACT**. Suppose in the training corpus, there is an instance *'I have a pet cat.'*, labeled with sense *cat/1*: this particular sense of *cat* falls into the class **NOUN.ANIMAL**. Since this is a candidate sense class for *dog*, the smallest sense number of *dog* which falls into class **NOUN.ANIMAL** is queried for; this happens to be sense 1. Then a straightforward lookup for Jiang and Conrath similarity between senses *cat/1* and *dog/1* returns a similarity value of 0.546. This is the weight of the example in the training set for word *dog*.

Testing with the validation set showed that it the examples were completely omitted from the training instance-base if they have similarity values to the test instance below a certain threshold, then the speed of the classifier can be dramatically improved, and performance could also be enhanced. Therefore a similarity weight threshold was introduced, which was adjusted using the validation data set.

A freely available implementation due to Ted Pedersen and colleagues (Pedersen, Patwardhan, and Michelizzi, 2004; Patwardhan, Banerjee, and Pedersen, 2003) was used for calculating similarity for all measures except the Jiang and Conrath measure. For the latter, we used our own implementation, based on SEMCOR corpus frequencies.

3.2.2 Similarity Measures

Several measures were tested for similarity values between instances, and the best performing was the Jiang and Conrath measure. Figure 2.7 shows how the animal words we discussed in section 3.2 are related to each other in this measure. A detailed discussion on the other similarity measures is given in section 2.3.1.

3.3 Voting

As mentioned earlier, three feature types were used in three separate classifiers, and the results were combined through voting, along with a 'classifier' output that always predicted the class belonging to WORDNET first sense of the word considered.

A discussion is provided in the next chapter on how invididual classifiers and voting results vary in performance at the evaluation experiments. Simple majority voting could improve the performance over the baseline, and a weighted majority voting could improve over this.

3.3.1 Weighted Majority Algorithm

The algorithm that was adapted for weighted majority voting is described in (Littlestone and Warmuth, 1994). The original algorithm is designed for binary classification problems, and the modified algorithm handles multiple classes, and works in the following way:

Initially, all classifiers have the same weight of 1. The classifiers participate in voting in a classifying experiment, on the set of instances in the development data set. Each classifier votes for its predicted class with its current weight. The weighted sum of the votes for each class is compared, and the class which accrued the highest weight is taken as the prediction of the algorithm. If this output is wrong, the weights of the classifiers which contributed with the wrong answer is decreased by a factor β such that $0 \leq \beta < 1$. The purpose of the validation experiment is to determine the value of β. The optimal value for β is determined by comparing the total number of accurate predictions the algorithm could make on development data.

Two levels of optimization is possible: First, β can be globally optimized for all words using the generic development data set. Second, the weights can be determined for individual words, using the development samples collected for individual words (described in section 3.1.1).

3.3.2 Compiling SENSEVAL Outputs

This system with WORDNET LFs can only be used for nouns and verbs, as there are no practically useful lexicographer files defined for adjectives and adverbs. However, in the final evaluation, it was necessary to include these parts of speech, for fair comparison with officially published results. So when compiling final SENSEVAL answers,

the baseline WORDNET first sense was used for these two parts of speech to fill in the results.

Additionally, SENSEVAL test data SGML files have some multiple-word phrases marked separately, in the manual annotation phase. When these instances can be identified as having a specific WORDNET sense entry (such as *'school board'*), this entry was used as the corresponding target sense for the whole phrase. Usually phrases have only one sense, and running a classifier on them does not make sense. Therefore WORDNET first sense was simply used to label these instances.

Some instances could not be correctly labeled with any of these methods (see section 4.2 for examples), and they are marked with 'U' (for unlabeled) as per the guidelines. In the evaluation, these cases were obviously marked wrong unless the human annotators also used U as the answer for some reason.

3.4 Support Vector Machine Implementation

One question that arises is whether the class system is strictly dependent on the k-nearest neighbor classifier we used. This is a reasonable doubt, as our systems heavily depend upon exemplar weighting, which has a very intuitive implementation with k-nn classifier.

State of the art WSD research has shown (Lee and Ng, 2002) that Support Vector Machines (**SVM**) (Vapnik, 1999) yield impressive performance on lexical sample task. To test the applicability of different systems, we implemented an alternative system using SVM as the classifier, while keeping the basic details of sense mapping and feature weighting intact. There are two major differences: First is the way features are represented, because SVM classifiers cannot handle nominal data. Second difference is about how example weighting is implemented in the classifier.

3.4.1 Feature Vectors

The set of features used are the same as in the earlier experiments; however there is the need to convert the features to binary as SVMs cannot handle nominal data. The

concatenated feature vector was converted into a binary vector. Cursory experiments with the development data showed that, unlike the case with k-nn classifier, a classifier that works on the concatenated vector of individual feature vectors, rather than a voted-combination of different classifiers, can perform better. This is in accordance with some previous observations as well; for instance, Lee, Ng, and Chia (2004) used the same classifier system used by Lee and Ng (2002) that reported good results on SENSEVAL-2 data, and could report third best performance reported in SENSEVAL-3 exercise. Their system used a single classifier and a combined feature vector.

The feature vectors were the same as earlier, with local context and part of speech were added as a window of up to three words to both sides. The combination was converted to a binary vector where a feature with n values was represented by a binary vector of n dimension; The value at the i^{th} position is 1 if the actual feature value is the i^{th} one in the ordered set of values. All other values in the binary vector are set to 0. Finally, vectors for all features were concatenated together. Classifier options were selected using the same set of validation instances (1000 instances for each part of speech) that were used in the previous experiments. SVM is a binary classifier by default; the multi-class classification is handled in one-against-one approach (Friedman, 1996). First, a number of binary classification problems are solved for all binary combinations of classes; then the classifier outputs vote for the final class. The class with the maximum number of votes is selected as the final class.

3.4.2 Example Weighting

Suppose we have a set of labeled training examples $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ and associated labels $y_1, y_2, y_3, \ldots, y_n \in \{-1, +1\}$. In the linearly separable case, we ideally want to find a separating hyperplane $\mathbf{x} \cdot \mathbf{w} + b = 0$ such that, for all labeled instances, the conditions

$$\begin{aligned} \mathbf{x}_i \cdot \mathbf{w} + b &\geq +1 \text{ whenever } y_i = +1, \text{and} \\ \mathbf{x}_i \cdot \mathbf{w} + b &\leq -1 \text{ whenever } y_i = -1. \end{aligned}$$

hold on the hyperplane. Equivalently, the hyperplane satisfies

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \qquad \forall\ i.$$

In support vector machines, we try to maximize the margin of distance from the separating hyperplane to the closest positive and negative examples. Simple algebraic geometry can show that this is equivalent to minimizing $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ subject to the constraints $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0,\ i = 1, 2, 3, \ldots, n$ in the linearly separable case.

In the non-separable case we introduce slack variables ξ_i for each instance such that $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \xi_i$, and try to minimize

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i, \qquad \xi_i > 0\ \forall i.$$

It is possible to introduce the example weights here, by penalizing the errors disproportionately; the errors in the instances with higher weights are counted with a stronger penalty. This can be done by changing the objective function to

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\sigma_i\xi_i\ ,$$

where σ_i is the example weight of the i^{th} instance. Effectively this adjustment in weights tells the optimization procedure that it must try to reduce the classifier error for those examples with high weights.

The implementation of weighted SVM classifier is based on LIBSVM (Chang and Lin, 2001), which employs C-support vector classification (Cortes and Vapnik, 1995; Vapnik, 1998).

3.5 Summary

Learning generic sense classes can borrow many techniques from the state of the art of WSD, but some issues have to be addressed with different and special techniques. This chapter provided a description of essential details behind the implementation of

a generic sense learner framework.

A system that learns generic word sense classes instead of fine grained senses is similar to a system used for conventional WSD, when it comes to features and representation. However, there are two major differences: First is how to reduce noise arising from training examples that come from different words, which may or may not be semantically similar to the word being classified. The second is how the fine grain senses are obtained once coarse grained classes are available as the classifier output.

We address the first issue by introducing a measure of *semantic similarity*, which we use to *weight* the training examples. The implementation of weights depends on the actual classifier. In a k-nearest neighbor classifier the weights are used to adjust the distances from the training instances to testing instance. With support vector machines, the optimization function is modified to penalize different examples' errors according to their 'importance', defined by their semantic relatedness to original word. The second problem is solved by the arbitration rule that WORDNET senses with smaller sense numbers have precedence over those with larger numbers. This heuristic is based on the fact that WORDNET senses come ordered in their corpus frequency in SEMCOR, and the assumption that this order is reasonably preserved in the test corpus.

Most of the implementation is generic, and does not rely on any special assumptions on the class map, other than the properties we discussed in sections 2.4 and 2.5. For demonstration purposes, WORDNET lexicographer files were used as the reference class map. The empirical results related to this particular class map are discussed in the next chapter. Apart from this specific detail, the rest of the implementation applies to the class maps we discuss in chapter 6 as well.

Chapter 4
Analysis of the Initial Results

The following sections will provide an analysis of results of the WSD system described in the previous chapter, which implemented a generic sense class learning framework using WORDNET lexicographer files as sense classes. Final evaluations were done on SENSEVAL-2 and 3 English all words task evaluation data.

For comparison purposes, the results will be shown for settings other than the optimal classifier settings (such as performance variation with different context window sizes) on SENSEVAL data sets, unless otherwise stated. The results reported such way are from the experiments that were run for comparison purposes only; the results from these experiments were not used and in setting the system parameters. The development data set (section 3.1.1) was used for this purpose.

All performance values reported were generated using the SENSEVAL official scorer whenever possible, and the values reported are recall values. Precision and Recall values are defined as

$$\text{precision} = \frac{\text{total number of instances correctly classified}}{\text{total number of instances attempted}}$$

$$\text{recall} = \frac{\text{total number of instances correctly classified}}{\text{total number of instances in the test set}}$$

The recall value is always smaller than the precision value for a given run, and provides a fairer comparison for an unrestricted task.

	SENSEVAL-2	SENSEVAL-3
Baseline (WORDNET first sense)	0.658	0.643
No substitute words (k-NN)	0.662	0.653
No substitute words (SVM)	0.663	0.652

Table 4.1: Combined baseline performance in SENSEVAL data for all parts of speech, including multiple-word phrases etc. 'Baseline' is the WORDNET first sense, and 'no substitute words' is the performance of a system trained only using labeled training examples from exact word as being classified.

4.1 Baseline Performance Levels

The standard baseline for SENSEVAL supervised WSD tasks is WORDNET first sense. Unfortunately, there were no official baseline score reported, and different systems had small differences in the figures they reported. For fair comparison with our system, the baseline figures reported here were calculated using the same classifier framework, except for the classifier component.

In addition, it is possible to have a baseline measure defined for our specific task. This is the performance of our classifier if it used only the examples that belong to the exact word being disambiguated, without using substitute examples from other words. This baseline helps measure the improvement the system gains by using examples from different words.

The combined baseline figures are given in table 4.1. This values include all parts of speech; all instances that are not classified (adjectives, adverbs, multiple-word phrases in WORDNET, and monosemous words) were labeled with their respective WORDNET first senses. These baseline values are also given in the tables where final SENSEVAL data are reported, (tables 4.4 and 4.5) for easy reference.

In the sections starting from section 4.3, a detailed analysis is provided on the results of k-NN classifier, which is our focus. To facilitate the component-wise analysis, the two baseline figures for nouns, verbs, and the total of nouns and verbs combined together are shown in tables 4.3 and 4.2, excluding other parts of speech which were not used in classification. These measures are the ones we are more interested in, as we will be focusing only on nouns and verbs. However this performance figures are different from the final system which includes adjectives, adverbs and multi-word phrases

	SENSEVAL-2			SENSEVAL-3		
	noun	verb	total	noun	verb	total
Baseline (WORDNET first sense)	0.711	0.439	0.618	0.700	0.534	0.626
No substitute words (k-NN)	0.719	0.435	0.622	0.712	0.549	0.639

Table 4.2: Baseline performance in SENSEVAL data for nouns and verbs. 'Baseline' is the WORDNET first sense, and 'no substitute words' is the performance of a system trained only using labeled training examples from exact word as being classified, using k-NN classifier. The 'total' value here is only over nouns and verbs.

	Development Data		
	noun	verb	total
Baseline (WORDNET first sense)	0.644	0.574	0.609
No substitute words (k-NN)	0.647	0.586	0.617

Table 4.3: Baseline performance in development data for nouns and verbs. Legend as per table 4.2.

which have distinct WORDNET entries.

4.2 SENSEVAL End task Performance

The final results were compiled by combining the results for nouns and verbs with the multi-word phrase results and filling in the rest of the instances with WORDNET first senses, as described in section 3.3.2.

As it was mentioned earlier, the baseline measure reported from different systems were different from each other. Our observations showed that even better figures are possible with better heuristics to identify multi-word phrases and errors in lemmas and parts of speech accurately. For instance, the instance `d001.s001.t013` in SENSEVAL-3 test data is '%', which does not have a WORDNET entry. The official answer key has it labeled as `percent%1:24:00::`. The sense tag is obviously non-trivial. Although this kind of instances can be expected in any practical text, they make comparisons among the systems hard, especially because the margin of improvement over the baseline is not large.

Tables 4.4 and 4.5 respectively show the final results for SENSEVAL-2 and SENSEVAL-3 data, using weighted majority voting for classifier combination. (Different voting schemes are compared in table 4.14).

System	Recall
Baseline (WORDNET first sense)	0.658
Baseline (No substitute words - k-NN)	0.662
Baseline (No substitute words - SVM)	0.663
SMUaw (Mihalcea, 2002)	0.690
Sense Classes: k-NN	**0.674**
Sense Classes: SVM	**0.670**
CNTS-Antwerp (Hoste, Kool, and Daelmans, 2001)	0.636

Table 4.4: Results for SENSEVAL-2 English all words data for all parts of speech and fine grained scoring.

System	Recall
Baseline (WORDNET first sense)	0.643
Baseline (No substitute words - k-NN)	0.653
Baseline (No substitute words - SVM)	0.652
Sense Classes: k-NN	**0.661**
Sense Classes: SVM	**0.659**
GAMBL-AW-S (Decadt et al., 2004)	0.652
SenseLearner (Mihalcea and Faruque, 2004)	0.646

Table 4.5: Results for SENSEVAL-3 English all words data for all parts of speech and fine grained scoring.

Each table shows the final results of our system, baseline figures for each test, and the results of the two systems that reported the best official results. Our systems results are shown as 'Sense classes', for both k-NN (TIMBL) classifier as well as the SVM classifier. Both k-NN and SVM based classifiers perform better than the reported state-of-the-art systems in SENSEVAL-3 and better than all but one system (that of Mihalcea (2002)) in SENSEVAL-2. The baseline figures also include two additional comparison measures: the performance of the systems trained with no substitute words (i.e. using labeled examples for the exact tested word). These results understandably differ for k-NN and SVM classifiers, so both systems are reported here.

In the following, we will use only the results of the k-NN classifier, which is our main focus; this classifier consistently gave better results than SVM.

Classifier	Development	SENSEVAL-2	SENSEVAL-3
Baseline (WFS)	0.609	0.618	0.626
Baseline (NSW)	0.617	0.622	0.639
POS	0.594	0.616	0.614
Local context	0.612	0.627	0.633
Syntactic Patterns	0.601	0.620	0.612
Combined, NFS	0.591	0.629	0.639
Concatenated	0.558	0.609	0.611
Combined (voting)	**0.625**	**0.631**	**0.643**

Table 4.6: Results of baseline (WFS: WORDNET first sense, NSW: supervised learning with no substitute words), individual, and combined classifiers: recall measures for nouns and verbs combined. Results given separately for nouns and verbs can be found in tables 4.7, 4.8, and 4.9.

4.3 Individual Classifier Performance

Performance on individual classifiers results are shown in table 4.6 along with the baselines. The entries 'POS', 'Local context' and 'Syntactic Patterns' are for classifiers that used the respective type of feature alone. 'Combined' is the final system where the separate classifier outputs were combined through voting.

'Combined, NFS' result is for weighted voting of the three classifiers without including the WORDNET first sense classifier. The listing shown as 'concatenated' is for a system that used a concatenated vector of the three types of features in a single classifier, instead of combining classifiers through voting. It is visible that combining features this way does not perform well. This is due to the fact that features with statistically weak yet useful information (such as syntactic patterns) can get ignored when competing with statistically strong features (such as part of speech).

It can also be seen that local context itself can outperform the baseline, and combining three features through voting can significantly increase the performance.

4.4 Contribution from Substitute Examples

The similarity values (for Jiang and Conrath measure) between an overwhelming majority of the sense pairs are below 0.5.

The average proportions of example instances that fall above different similarity

weight threshold values are shown in figure 4.1. Averages were calculated for all words in the development data set, and the similarity measure used was Jiang and Conrath. The average number of all training examples for a word is about 15,000 for nouns, and about 24,000 for verbs. The large number of examples for verbs is partly due to the fact that they have a fewer number of lexicographer files; this makes the chance of a random sense falling into a given lexicographer file is higher than the same chance for nouns. The total number of training instances collected for all nouns and verbs is almost equal for both parts of speech (21,984,292 and 21,296,351 respectively).

It is interesting to note that the similarity distribution characteristic for both nouns and verbs are strikingly similar to each other. This is not intuitive because noun and verb hierarchies are significantly different from each other, in terms of depth, number of children per node, and also in terms of information content —as the frequencies of senses are fairly different from each other.

The substitute examples contribute to the classifier performance in somewhat expected way: when the instances keep getting added according to their similarity (starting from the most similar instances), the performance gets increased up to certain point, after which the performance starts to drop with adding more instances. The initial improvement of the performance is due to the fact that newly added instances yield information that were not available from the labeled training instances of the original word. As the similarity threshold drops, the new instances start introducing unrelated features which end up as noise; At a certain point, the noise overcomes the information, and performance starts to drop from there.

Figure 4.2 shows the variation of classifier performance on development data, for nouns and verbs, against the similarity weight threshold that was used to filter the training instances. (Instances that have similarity weights lower than the threshold are removed from the training set.) As the threshold increases, the performance reaches the level of performance with only the original word instances as training set. When the threshold decreases, the performance can go *below* this margin, as the new instances can contribute to noise.

It could be observed (in section 4.3) that the syntactic pattern feature performs rea-

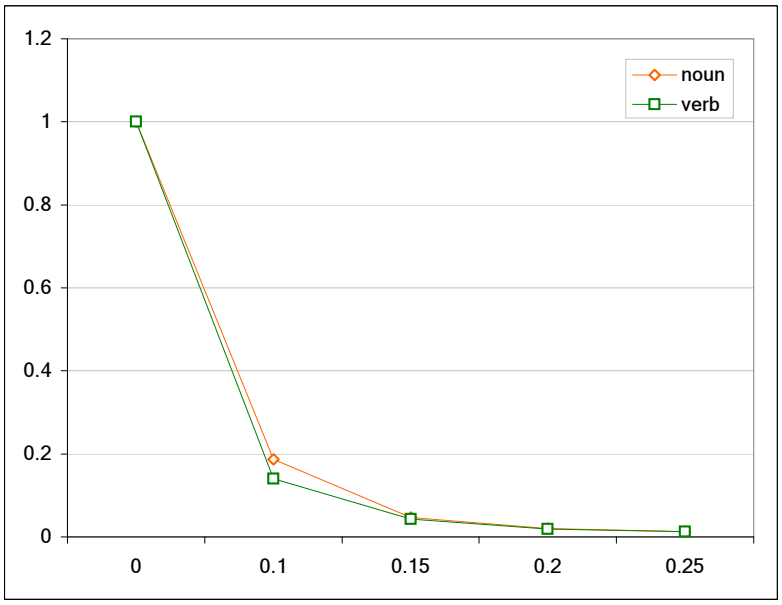

Figure 4.1: Average proportions is instances (over all words) that fall above a given weight threshold. Nouns and verbs show strikingly similar patterns, despite much different sense distributions and WORDNET hierarchy depths.

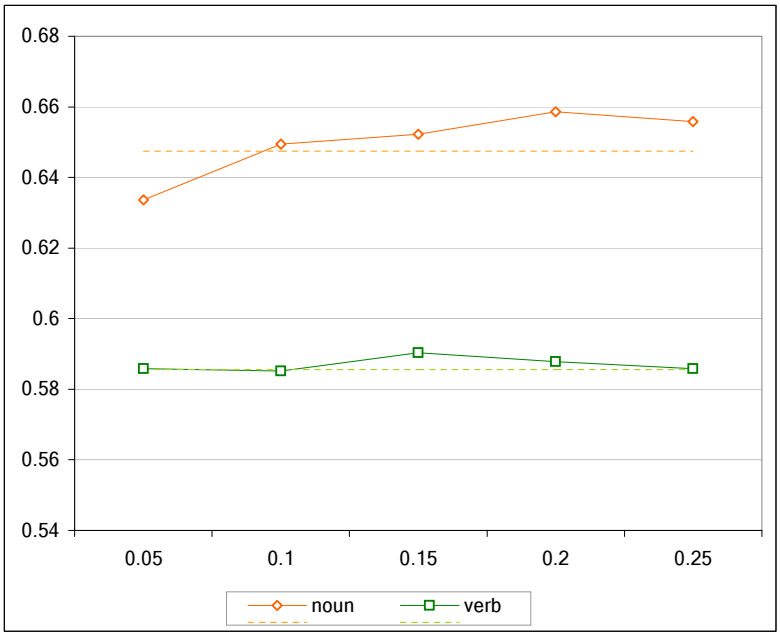

Figure 4.2: Variation of classifier performance with new examples. Training instances are filtered using similarity weight (x-axis) as the threshold. Dotted lines show the level of performance with only the instances of exact word. The performance increases with more similar instances added up to a certain point, and then starts to drop again, reaching the performance of a system that uses only original word examples.

	same word		**similar words**	
	noun	verb	noun	verb
POS	0.617	0.570	0.611	0.570
local context	0.627	0.576	0.647	0.571
patterns	0.635	0.548	0.645	0.558
combined	**0.647**	**0.586**	**0.659**	**0.590**

Table 4.7: Comparison of performance using same-word and substitute-word examples: development data

	same word		**similar words**	
	noun	verb	noun	verb
POS	0.712	0.439	0.704	0.448
local context	0.724	0.440	0.724	0.439
patterns	0.709	0.435	0.716	0.435
combined	**0.719**	**0.435**	**0.724**	**0.455**

Table 4.8: Comparison of performance using same-word and substitute-word examples: SENSEVAL-2 English all-words task data

sonably well, even though it is much sparser compared to part of speech and local context features when individual words are considered. This raises the question whether the pattern feature specifically benefits from substitute examples. Tables 4.8 and 4.9 show the comparative results for the two classifier systems side by side. First uses only labeled examples from the same word as training instances, and the other, the system we are currently discussing.

Interestingly, part of speech classifier works better without substitute training instances in most cases. Local context classifier performs almost similarly, with only minor variations. However, in both cases the answers correctly classified are not essentially the same; this fact determines how the classifiers behave on the final combination. In the case of patterns, this trend reverses, and substitute words clearly add to the clas-

	same word		**similar words**	
	noun	verb	noun	verb
POS	0.702	0.542	0.688	0.520
local context	0.709	0.551	0.706	0.542
patterns	0.672	0.515	0.684	0.522
combined	**0.712**	**0.548**	**0.719**	**0.548**

Table 4.9: Comparison of performance using same-word and substitute-word examples: SENSEVAL-3 English all-words task data

sifier performance.

It is hard to explain exactly why the patterns benefit more from substitute words; there can be more than one possible reasons for this:

First, the pattern feature is sparse compared to the part of speech and local context features. Since the number of possible values at a given position is rather small for part of speech feature (typically 30-35), even the same-word example setting could be providing enough clues for the classifier to learn. Local context feature is moderately dense; the number of words that immediately surround a given word is determined, to some extent, by grammatical constraints: Common tokens such as propositions play a significant role. (Later in section 6.7.4 we show that the feature information gain ratio for immediate neighbors of the word are typically high compared to the words far away in the context.) Compared to these two features, the syntactic pattern feature is much sparse, and most instances do not have any values selected for the syntactic pattern feature. Substitute words are meant to alleviate exactly this issue, by providing more examples, albeit of lesser quality. So the same instances that can harm the overall quality of POS and local context data sets can provide useful clues for patterns.

Another possible reason is somewhat related to the importance of information given by pattern-like features, as also shown in previous work, such as selectional constraints based WSD (Resnik, 1996). Syntactic patterns (see section 3.1.3) capture information similar to selectional constraints, and they may be inherently stronger in capturing class-based generalizations. For instance, a selectional constraint predicate in the style `beverage(X) ← object-of(drink, X)` can be directly captured by the object-of feature in the syntectic patterns feature set.

4.5 Effect of Similarity Measure on Performance

The effect of different similarity measures on the result is shown in table 4.10. 'No substitute words' line show the baseline performance of the system if we only used the examples from the word that is being classified, without using any substitute word examples from different words. 'No similarity' is the scheme where substitute examples

	SENSEVAL-2	SENSEVAL-3
Baseline (WFS)	0.618	0.626
Baseline (NSW)	0.622	0.639
No similarity used	0.608	0.599
Lesk	0.610	0.613
Resnik	0.540	0.522
Lin	0.590	0.586
Hirst and St. Onge	0.581	0.606
Jiang and Conrath	**0.631**	**0.643**

Table 4.10: Effect of different similarity schemes on recall, combined results for nouns and verbs. for comparison, baseline (WFS: WORDNET first sense; NSW: no substitute words) are provided.

were used, but similarity voting was not used (thus considering the substitute words as being equally authoritative as the original word). Weighted schemes include Lesk, Lesk (1986), Resnik (1995), Lin (1997), Hirst and St-Onge (1998), and Jiang and Conrath (1997). The Jiang and Conrath measure performs the best, and some weighting systems perform worse than the baseline. The fact that the most complex representation was used in Hirst and St. Onge does not seem to increase the performance of that similarity measure.

4.6 Effect of Context Window Size

One fact observed in the experiments was that the system does not depend on a wide window of collocations; this may be due to the fact that the wider the window, the more different the behavior of words from each other. This is not a stand-alone observation: Mihalcea, for instance, reported fairly good improvements in large-scale sense tagging with a one-word window to both sides (Mihalcea and Faruque, 2004). Some of the lexicography-oriented approaches for sense disambiguation, such as WASPS and WORD SKETCH systems (Kilgarriff and Tugwell, 2001a; Kilgarriff and Tugwell, 2001b) seemingly support this idea, by stressing their focus on local features. Their set of relations are somewhat similar to the syntactic-pattern based features in this work, except for the fact that the relations they presented to the annotators were filtered for statis-

Part of Speech	Window Size -1, +1	-2, +2	-3, +3
nouns	0.645	0.659	0.644
verbs	0.579	0.590	0.583
all	**0.612**	**0.625**	**0.614**

Table 4.11: Performance of the system with different sizes of local context window in development data.

Part of Speech	Window Size -1, +1	-2, +2	-3, +3
nouns	0.724	0.724	0.712
verbs	0.446	0.455	0.458
all	**0.629**	**0.631**	**0.626**

Table 4.12: Performance of the system with different sizes of local context window in SENSEVAL-2 data.

tical significance in the British National Corpus[1]. Several other researchers reported similar observations, favorable for a narrow context window (Lesk, 1986; Martinez and Agirre, 2000). Choueka and Lusignan (1985) also claimed that humans can disambiguate a great majority of words using typically two-word context window to each side of the word.

Behavior of the system for different sizes of local context window is shown in tables 4.11, 4.12, and 4.13. All the other parameters were kept at their optimal settings while the context window size was varied.

Experiments with Validation data reported best performance at context window size [-2, +2] for both nouns and verbs, so this window size was selected in the final results. The optimal results for SENSEVAL data looks slightly different in the case for verbs. In both test cases, verbs show best results at [-3, +3] window; the performance

[1]British National Corpus is a 100-million representative selection of contemporary British English, both spoken and written. http://www.natcorp.ox.ac.uk/

Part of Speech	Window Size -1, +1	-2, +2	-3, +3
nouns	0.699	0.719	0.714
verbs	0.541	0.548	0.551
all	**0.629**	**0.643**	**0.641**

Table 4.13: Performance of the system with different sizes of local context window in SENSEVAL-3 data.

differences in verbs for the two cases in SENSEVAL data are not significant though, ranging at about 0.3%. In addition, as it seems from noun results as well, it may be the case that the size of context window does not have a consistent relationship with performance, and [-2, +2] would have been a 'safe guess'.

In our case, the most important reason for the success of the small context window might have been the representation of semantic-class based word senses; In manual observations it was seen that most of the common features that discriminate classes are available within a narrow context. For instance, if we consider Resnik (1996) style selectional constraints, the subject-verb and verb-object relationships, though not always, occur reasonably frequently within a narrow context window. Even when they do not, the syntactic pattern feature can be compensating for them, as it is not dependent on the context size.

This kind of short-window context can be thought of as being less sensitive to the genre variations, because most of the basic connectors of the grammar within a narrow context window do not change over genre, while genre can significantly impact the wider 'topical' context. Also it's easier to find more *different* narrow-window examples from a limited corpus such as SEMCOR using the techniques we used here; if we were to rely on topical context, we would have to have a lesser number of distinct examples, because all the words in a given document share almost the same wide-window context, with the exception of the particular word itself being removed.

4.7 Effects of Voting

Table 4.14 shows the performance increase for multi-class words in SENSEVAL data sets with different voting schemes. It should be noted that the increase of performance is not very high, and not statistically significant. However, given lower performance improvements over baseline in the state of the art systems, even an improvement of this magnitude is considerable. The validation sets for SENSEVAL-2 and SENSEVAL-3 (considering the weight adjustments per individual words) had 5.8% and 6.2% performance increases respectively. This has been a common observation in English all-words task

SENSEVAL-2	nouns	verbs	total
Baseline	0.711	0.439	0.618
Simple majority	0.724	0.455	0.631
Global Weights	0.728	0.453	0.634
Individual Weights	0.740	0.453	0.642

SENSEVAL-3	nouns	verbs	total
Baseline	0.700	0.534	0.626
Simple majority	0.719	0.548	0.643
Global Weights	0.728	0.549	0.649
Individual Weights	0.728	0.552	0.650

Table 4.14: Improvements of recall values by weighted voting for SENSEVAL English all-words task data.

system evaluations. For instance, (Decadt et al., 2004) report 8% performance loss between validation and testing data (from 12% to 4%), and (Hoste et al., 2002) reported a 20% accuracy drop.

In a way, this supports our arguments (section 1.3.1) that optimization techniques typical for classic lexical-sample style WSD problems are not robust enough to perform well in the extreme low-data conditions. Nevertheless, it shows that small improvements are possible even with this kind of conditions, with proper algorithms.

The weighting algorithm, especially the one that works at word level, has a significant bias towards most frequent sense. This is a desirable feature in a way, because of the strong skewness of the sense distribution; accuracy of a classifier that always labels senses with WORDNET first sense is fairly high; deviating from WORDNET first sense is a risky decision, unless the classifier is very confident when it guesses a sense to be something other than (the most likely) first sense. The voting algorithm provides this confidence by strongly biasing the classification towards the first sense, as the first sense has a higher accuracy rate on development data.

In the actual test data sets this effect was obvious; Only 14 of all SENSEVAL-2 nouns had been marked as a sense other than WORDNET first sense, when the correct answer was the first sense. For verbs, this figure is only 4. In SENSEVAL-3 data set, only 10 of the nouns and 6 of the verbs were marked out of WORDNET first sense when the actual answer was the first sense. This shows that the classifier is fairly accurate when decid-

		SENSEVAL-2		SENSEVAL-3	
		fine	coarse	fine	coarse
noun	baseline	0.711	0.790	0.700	0.781
	system	0.740	0.817	0.728	0.813
verb	baseline	0.439	0.713	0.534	0.781
	system	0.453	0.718	0.552	0.794

Table 4.15: Fine and coarse grained results compared, for SENSEVAL data.

ing against the first sense; this also contributes to the significance of the performance improvement, when a sign test such as McNemar test is used.

4.8 Error Analysis

Coarse grained (at lexicographer file level) results for the SENSEVAL-2 and SENSEVAL-3 data are shown, comapred with fine-grained results, in table 4.15. Baseline figures reported are for the lexicographer file occupied by the WORDNET first sense.

The improvement of performance over the baseline at lexicographer file level is not any larger than the performance improvement at fine-grained sense level. Although this looks somewhat counter-intuitive, the reason is purely a statistical phenomenon. Recall that when a fine grained classifier output is correct when the baseline is wrong, this output always falls in to a lexicographer file other than the one the first sense belongs to. As it is shown later in tables 4.16 and 4.17, the errors due to the sense loss (defined in section 2.5) are minimal; in this particular data sets, most of these errors happen when the correct secondary sense falls into the LF which has WORDNET first sense as its primary sense. At LF level, they increase the accuracy of both baseline and the supervised classifier by roughly the same number of correct instances. Addition of the same number of correct instances to both classifiers results in a reduction of percentage gain.

4.8.1 Sense Loss

There are some errors due to sense loss (see 2.5), although this is not the major source of errors. Even when our classifiers are accurate at class level, the fact that we can accom-

	SEMCOR		SENSEVAL-2			SENSEVAL-3	
lex file	coverage	hits	misses	coverage	hits	misses	coverage
act	**79%**	61	11	**85%**	59	11	**84%**
animal	**99%**	1	0	**100%**	10	0	**100%**
artifact	**89%**	74	3	**96%**	98	14	**88%**
attribute	**89%**	17	1	**94%**	17	4	**81%**
body	**92%**	119	5	**96%**	33	1	**97%**
cognition	**86%**	72	11	**87%**	38	8	**83%**
communication	**81%**	30	3	**91%**	27	9	**75%**
event	**93%**	12	5	**71%**	46	0	**100%**
feeling	**93%**	3	0	**100%**	9	0	**100%**
food	**78%**	1	0	**100%**	12	0	**100%**
group	**84%**	52	13	**80%**	32	0	**100%**
location	**87%**	20	0	**100%**	36	3	**92%**
motive	**99%**	0	0	**-**	3	0	**100%**
object	**91%**	0	0	**-**	5	0	**100%**
person	**81%**	142	20	**88%**	126	13	**91%**
phenomenon	**99%**	3	0	**100%**	2	0	**100%**
plant	**93%**	0	0	**-**	1	0	**100%**
possession	**90%**	3	1	**75%**	16	1	**94%**
process	**92%**	7	2	**78%**	1	0	**100%**
quantity	**97%**	10	0	**100%**	8	0	**100%**
relation	**99%**	1	0	**100%**	8	0	**100%**
shape	**96%**	0	1	**0%**	2	1	**67%**
state	**87%**	98	0	**100%**	22	2	**92%**
substance	**97%**	10	0	**100%**	2	1	**67%**
time	**74%**	41	6	**87%**	36	8	**82%**
TOTAL	**85%**	777	82	**90%**	649	76	**90%**

Table 4.16: Errors due to sense loss in nouns.

modate only one sense per a given lexicographer file means that some answers can still be marked wrong. We analyzed the theoretical impact of this in section 1.3.2, using a hypothetical classifier, and now we are in a position to measure the performance drop with an actual classifier.

Tables 4.16 and 4.17 show the sense losses per each lexicographer file, for nouns and verbs respectively. The column SEMCOR shows the sense loss calculated for SEMCOR corpus per each lexicographer file, and the SENSEVAL columns show the loss for SENSEVAL answer files for *actual classifier outputs*, as opposed to the hypothetical coarse-grained classifier with 100% accuracy we discussed in section 1.3.2. Only the instances that were correctly put into a class found in the official answer key are considered here:

	SEMCOR	SENSEVAL-2			SENSEVAL-3		
lex file	coverage	hits	misses	coverage	hits	misses	coverage
body	**90%**	2	1	**67%**	14	1	**93%**
change	**81%**	37	14	**73%**	32	6	**84%**
cognition	**75%**	39	20	**66%**	39	10	**80%**
communication	**70%**	38	39	**49%**	42	28	**60%**
competition	**87%**	0	1	**0%**	1	0	**100%**
consumption	**90%**	8	0	**100%**	3	2	**60%**
contact	**71%**	7	8	**47%**	22	6	**79%**
creation	**82%**	12	2	**86%**	1	1	**50%**
emotion	**89%**	7	1	**88%**	23	3	**88%**
motion	**77%**	6	5	**55%**	44	23	**66%**
perception	**82%**	15	10	**60%**	33	21	**61%**
possession	**79%**	10	9	**53%**	22	7	**76%**
social	**85%**	13	6	**68%**	21	10	**68%**
stative	**71%**	57	30	**66%**	107	59	**64%**
weather	**91%**	0	0	**-**	0	0	**-**
TOTAL	**77%**	**251**	**146**	**63%**	**404**	**177**	**70%**

Table 4.17: Errors due to sense loss in verbs.

'hits' are the number of instances that were put into the correct fine-grained sense (i.e. the cases where the primary sense of the class was also the correct answer), and 'misses' are the cases where the classifier was correct on the output class, but the actual answer in official answer key was a secondary sense within the class. These errors cannot be avoided by increasing the classifier accuracy, as they result from the class map alone. The columns 'coverage' show the percentage of correctly classified instances that could make it to correct answers, being the primary senses per respective classes.

One obvious fact is that there is no significant difference of losses in sense loss between different classes of the same part of speech. Nouns yield about 90% coverage (10% sense loss) while verb coverage is about 70% (with 30% loss). The fact that different classes have different loss patterns in SEMCOR and SENSEVAL data, and the fact that the losses for all classes are generally in the same order as the average loss for that part of speech, show that the issue is not something that can be fixed at individual class-level. Also, the smaller number of classes in verbs can easily explain the higher loss thereof.

These issues lead to the conclusion that the most promising way of reducing the sense loss is to increase the number of classes and, in particular, try to reduce the num-

ber of secondary senses that fall within each class. In other words, the only practical way to reduce sense loss is to design a set of classes that splits fine grained senses of a given word in such a way that the chance of two senses falling into the same sense class is minimal. In the next two chapters, we try to address this issue.

Confusion matrices for the classifier at class level are shown in tables 4.18, 4.19, 4.20 and 4.21.

Confusion matrices for nouns are much cleaner than those for verbs, with a stronger diagonal component. This may be partly due to the greater number of classes that makes it possible, at least theoretically, to have better cohesion within a given class. If this is true, it supports our previous argument for a larger number of classes with comparatively fewer senses in each. In addition, the inherent vague nature of verb classification would have played a part in larger confusion among verb classes.

In both parts of speech, confusion among classes seem to be mutual in general. That is, if class *a* gets confused by the classifier as class *b*, then there is a good chance of predicting class *a* in place of class *b* as well.

Similarly, there are a few classes that lead to confusion more than others; for instance, classes **ACT**, **COGNITION**, and **STATE** in nouns show major confusions. These two classes occupy in average only 8%, 4% and 3% proportions on the training corpus; so the likely source of confusion is not related to their statistical power, but their feature vectors. **BODY** is the class that has cleanest defining features as it is evident from both SENSEVAL tests. The percentages of instances of class **BODY** marked as other classes in SENSEVAL-2 and SENSEVAL-3 are 2% and 6% respectively. Similarly, there are no instances in SENSEVAL-2 that were marked incorrectly as belonging to class **BODY**; this is very significant when observing the fact that there are more than 120 instances correctly classified into the class. In SENSEVAL-3, five instances got incorrectly classified into **BODY**.

In verbs, there does not seem to be any significant patterns, and the lesser emphasis on diagonal component (compared to the nouns) with respect to all classes hints that there are no specific 'problem classes'. As we showed in figure 4.1, the average number of training examples per class remain mostly the same for both examples. Also, as the

	act	animal	artifact	attribute	body	cognition	communication	event	feeling	food	group	location	motive	object	person	phenomenon	plant	possession	process	quantity	relation	shape	state	substance	time
act	75			4	1	7					2			2		2			4		1		2		
animal		1													1										
artifact			64			2					1														1
attribute				19		4		1							1										1
body					122																				
cognition				8		81	3	1			5	1		1						1					
communication				1			3	1				1		3											
event							32	16			3	20	1			3									
feeling							1		3																
food										1															
group			10				1				60	2		1	3			2					1		
location				1			2					20													
motive							5																		
object			3								2												1		
person						1								2	162										1
phenomenon								1								4				1					
plant																									
possession				1		1									1			4							
process											2								8				1		1
quantity					1									2		1			8						
relation																				1					
shape																					1				
state	3			1				1							1							1	97		1
substance																							9		
time								1						2	1				1						45

Table 4.18: Confusion matrix for SENSEVAL-2 nouns. Original classes in rows, classifier output in columns.

	act	animal	artifact	attribute	body	cognition	communication	event	feeling	food	group	location	motive	object	person	phenomenon	plant	possession	process	quantity	relation	shape	state	substance	time
act	68		2			5		4		1								1		1					
animal		9																							
artifact			111		1	1	2			7					1							2		2	
attribute				24				1														5			
body					34						1														
cognition						36	1	2							4										
communication							38								1										
event								45	1																
feeling									9																
food										11															
group			7								32														
location												38													
motive													3												
object														5											
person			1			4	1							1	136		1								
phenomenon																3									
plant																	1								
possession																		17		1					
process																			1						
quantity	1																			8					
relation																					9				
shape											1											3			
state			2	5		1		1			4				1								24		1
substance			2									1		1						1				3	
time															2								1		40

Table 4.19: Confusion matrix for SENSEVAL-3 nouns. Original classes in rows, classifier output in columns.

	body	change	cognition	communication	competition	consumption	contact	creation	emotion	motion	perception	possession	social	stative	weather
body	3	1						2				2		1	
change		52	1	1	1		1			5	1	3	4	5	
cognition		2	58	3			1		1	2	6	1	1	3	
communication		4	1	78			1	1	1	1	3	2	2	1	
competition					1					1				1	
consumption						7			1					2	
contact			1	1			15						2		
creation			2	5	1			12			4	4	8		
emotion				1			2		9	1					
motion		2	1				2			10		1	1		
perception			1	4			1				23		1		
possession			1									19	1	2	
social		1	1	1	1					3		1	17	2	
stative		5		3			3	3	1	6	2	1	1	95	
weather															

Table 4.20: Confusion matrix for SENSEVAL-2 verbs. Original classes in rows, classifier output in columns.

	body	change	cognition	communication	competition	consumption	contact	creation	emotion	motion	perception	possession	social	stative	weather
body	15			1					1	2	1				
change		37		1			1			9	2	4	3	1	1
cognition			49	1					2	1	3	1	1	4	
communication	1	2	5	69			1			3	3	1	3	1	
competition					1		1			2				1	
consumption					1	5									
contact				2			27	1		3		1			
creation								2					10		
emotion	2	1							26		1				
motion	2	5		3			1		1	65		1	2	1	
perception		1		1					3		54		2		
possession		2		3	1			2		1		29	4	3	
social				1			2			2	2		30	3	
stative		2	2	3						6	1	8		164	
weather		1													

Table 4.21: Confusion matrix for SENSEVAL-3 verbs. Original classes in rows, classifier output in columns.

	senses		LFs	
	nouns	**verbs**	**nouns**	**verbs**
simple average	1.23	2.18	1.13	1.59
SEMCOR instances	2.14	3.09	1.65	2.01
weighted average	6.63	11.99	3.82	4.45

Table 4.22: Average polysemy in nouns and verbs in WORDNET . Numbers shown are average number of senses.

	senses		LFs	
	nouns	**verbs**	**nouns**	**verbs**
simple average	0.240	0.513	0.157	0.289
weighted average	1.253	1.657	0.640	0.479

Table 4.23: Average sense entropy values for nouns and verbs in SEMCOR .

figure 2.3 shows, the average number of classes per number of senses for both nouns and verbs is also close to each other. For this reasons, it can be assumed that the higher confusion in case of verbs is due to an intrinsic difficulty of learning verb classes; this conclusion is in accordance with the observations of other people on the same data sets, such as (Hoste et al., 2002) and (Mihalcea and Faruque, 2004). Verbs, in general, are 'harder to learn'.

One possible reason for this difficulty can be that verbs have a higher average polysemy. Table 4.22 shows the average level of polysemy for nouns and verbs, in WORDNET 1.7.1 and SEMCOR corpus; 'simple average' is the average over all WORDNET senses, (excluding WORDNET entries for numbers, as they are not proper English words). But this number is skewed due to the fact that there is a large number of monosemous words that never appear on our labeled corpus, i.e. SEMCOR. For this reason, we calculate the averages over words that appear at least once in SEMCOR as well, which is shown in the second row as 'SEMCOR instances'. The third 'weighted average' line goes even further by weighting the word sense count by the relative frequency of the word in the corpus. This is a much higher value for all cases, confirming again the fact that more frequent words tend to be more polysemous as well. 'Senses' and 'LFs' show the figures in terms of fine grained senses and WORDNET lexicographer files, which were the sense classes for our classifiers.

Kilgarriff (2000) studies the polysemy counts and entropy measures of nouns and

verbs, and suggests that the sense entropy is a better than the polysemy count as a measure of difficulty in classification. Table 4.23 shows the average entropy for words in SEMCOR corpus. Entropy is calculated using the usual formula $-\sum p(s) \log p(s)$ over all senses s of a word. 'Simple average' row shows the average entropy over all words which were present in the corpus, while 'weighted average' shows the average weighted by the frequency of each word in the corpus. Results, as in table 4.22, are shown both for fine grained senses and for lexicographer files. The values for fine grained senses show patterns similar to those reported by Hoste et al. (2002). Our values tend to be in the lower-side as our average includes words that are monosemous. Hoste and colleagues reported the scores for a smaller sample of words, which contained only polysemous words; this leads to higher values for both sense count and entropy.

Unfortunately, a direct comparison between entropy values of nouns and verbs at lexicographer file level is not feasible as the number of LFs in the two systems are different, and smaller number of LFs in verbs can make the entropy values thereof to be lower.

4.9 Support Vector Machine Implementation Results

Final results of the support vector machine based system is shown in table 4.24. For easy comparison, the baseline results, the results of the classifiers trained without substitute examples, and the performance of the k-NN classifier are also shown. These are essentially the same as those reported in tables 4.4 and 4.5.

Table 4.25 shows the results for individual parts of speech, together with the respective baselines. The results are shown for nouns and verbs separately and nouns and verbs in total; 'No substitute words' baseline is the one with SVM classifier.

As we mentioned with the implementation details, the classifier that combined all features together in one vector performed the best. For this setting, weighted voting scheme is not applicable: However it seems that the performance is comparable with the corresponding weighted voting scheme with k-NN classifier. This augurs well with

SENSEVAL-2	Recall
Baseline (WORDNET first sense)	0.658
No substitute words (k-NN)	0.662
No substitute words (SVM)	0.663
Sense Classes: k-NN	**0.674**
Sense Classes: SVM	**0.670**

SENSEVAL-3	Recall
Baseline (WORDNET first sense)	0.643
No substitute words (k-NN)	0.653
No substitute words (SVM)	0.652
Sense Classes: k-NN	**0.661**
Sense Classes: SVM	**0.659**

Table 4.24: SVM classifier results for SENSEVAL English all words task data. Final result for all parts of speech and fine grained scoring.

SENSEVAL-2	nouns	verbs	total
Baseline	0.711	0.439	0.618
No substitute words	0.721	0.442	0.626
SVM	**0.734**	**0.449**	**0.637**

SENSEVAL-3	nouns	verbs	total
Baseline	0.700	0.534	0.626
No substitute words	0.713	0.542	0.637
SVM	**0.728**	**0.545**	**0.646**

Table 4.25: SVM-based system results. Final SENSEVAL results for all parts of speech was given in tables 4.4 and 4.5.

the generally agreed idea that k-NN classifier is more susceptible to large numbers of features, unlike the SVM classifier. In this particular implementation, this is partly due to the fact that the best performing feature selection scheme we found for our k-NN classifier —information gain ratio– is strongly biased against features with larger numbers of values, and incorrectly selects part of speech features as better when the local context features are much informative in reality. The $[-3, +3]$ word feature window of local context, for instance, contains about 42,000 (in total for all six positions) different values in our training data set, while the same size part of speech feature only has about 180 values (in total). This means that when put together, local context features are at a heavy disadvantage on information gain ratio feature weighting, which is essentially inversely proportional to the number of possible values of a feature.

4.10 Summary

We presented the performance figures of a proof-of-concept coarse grained sense learner prototype, using WORDNET lexicographer files as generic sense classes. It can be seen that even at the much coarse level of granularity that WORDNET LFs provide, it is possible to effectively learn the LFs as sense classes, and that the results can be practically useful. This is a part of the contribution of the core ideas presented in this thesis.

It must be noted that no argument was made, in the work that is presented so far, for the suitability of WORDNET LFs for this purpose. In the next chapter, we will focus on this question, and discuss what practical problems and inherent issues in LF design hinder the performance of generic class learning.

Chapter 5

Practical Issues with WordNet Lexicographer Files

The previous chapter discussed a prototype system, which demonstrated the feasibility of our proposal of learning generic sense classes using a pool of senses from different words. We also discussed how the noise due to disparities among senses can be reduced, using a proper example weighting scheme.

Recall that the sense map used in the experiment was directly derived from WordNet lexicographer files; this is a manually created system, taking into consideration the semantics of words and their *taxonomical* relations. Contextual features of the senses in text were not used, at least not explicitly. This raises the question whether similarities among usage patterns of different words conform to their proximity within WordNet hierarchical organization.

A simple answer to this question is that there is no confirmed evidence that they do. In fact, work of some linguists seem to suggest that the WordNet class structure does *not* conform to linguistic usage at least in some parts of the hierarchy. On the other hand, WordNet creators do not claim, in the first place, lexicographer files' fitness for use as generic classes for WSD. They make sense semantically, but the fact whether they are cohesive in terms of features that can be found automatically within text is not necessarily established. Although we reported satisfactory results using WordNet LFs as generic classes this was not proof that they are the best for the task.

In this chapter, we will discuss arguments from linguists and from our own obser-

vations regarding this issue. We will focus on the application at hand — learning the classes effectively from a set of labeled examples, and using the classes in fine grained WSD. We shall show what kind of problems with lexicographer files can impede the performance of our classifier.

Finally, the chapter suggests that creating a set of classes based on the features found within text itself can solve several of these problems.

5.1 Dogs and Cats: Pets vs Carnivorous Mammals

Deriving substitute training examples using WORDNET hierarchical information is nothing new; Leacock, Miller, and Chodorow (1998) used monosemous words found in WORDNET to derive training examples for polysemous words. SENSEVAL exercises had numerous systems that use WORDNET relatives as substitute examples in different ways; some examples for this approach are (Mihalcea and Faruque, 2004; O'Hara et al., 2004; Seo, Rim, and Kim, 2004; Agirre and Martinez, 2004; Agirre and Lopez de Lacalle Lekuona, 2004), and (Férnandez-Amorós, 2004). Although the implementations can differ, the general idea behind them can be explained by one approach described in what follows:

Each sense of a polysemous word has a synset, or a set of senses (from words other than the original word) that are synonymous with the original sense. One or many of the elements in the synset can be the only sense of a monosemous word. For instance, the word *teacher* has two senses in WORDNET :

1. a person whose occupation is teaching

2. a personified abstraction that teaches

The synset to which the sense 1 of this belongs is {*teacher/1*, *instructor/1*}. The synonym *instructor/1* is the only sense of *instructor*. As a result, we can take any unlabeled instance of *instructor* as being in the same synset as the first sense of *teacher*. Assuming we can substitute synonymous instances as examples for a given sense, the monosemous synonym *instructor/1* provides an easy way of gathering 'labeled' examples for

the first sense of *teacher*. This is not limited to senses in synsets only; one can extend the same idea for the neighbors in the WORDNET hierarchy as well, by using hierarchical neighbors when direct synonyms are not found.

One can think of the WSD framework presented in this work as working on a similar, but sophisticated technique, as Jiang and Conrath scheme ensures some level of hierarchical localization. While the Jiang and Conrath measure is dependent on the hierarchy, it provides somewhat better approximations for similarity as well, as demonstrated in other experiments (Jiang and Conrath, 1997). This can be expected to eliminate the noisy examples that can result in by using broad generic classes that include a large variety of senses (such as words *dog*, *cat*, *horse*, and *shark* for **ANIMAL** as shown in section 2.3.1). One problematic issue here is that this localization heuristic is not always sound in terms of linguistic usage. Taxonomical similarity measures do not address the fundamental problem of some senses being unsuitable as substitute examples for a particular sense, due to different contextual semantics.

Looking at the example (figure 2.7) it can be seen that *cat* and *dog*, both in **ANIMAL** sense, share the common parent *carnivore*. Consequently, the similarity values of carnivore to both *cat* and *dog* are greater than the similarity between *cat* and *dog*, in most similarity measures we discussed. These values are, for instance,

carnivore - cat	0.7567
carnivore - dog	1.9576
cat - dog	0.5457

for Jiang and Conrath measure. Path and Resnik measures, among many others, produce similar results.

It can be thought that since *carnivore* is a closer neighbor of *dog*, it can provide better examples for *dog* than *cat* could. But in practice, this is not the case; one can understand this considering, for instance, commonly occurring collocations such as *dog food – cat food*, *dog lover – cat lover*, or *pet dog – pet cat*, or co-occurrence with words such as *TV*, *toy*, *kids* in a wide context. In contemporary English, it is much likely that *dog* and *cat* are used in similar contexts —as household pets— compared to the word *carnivore*.

	50 words	**1000 words**
cat - carnivore	0.4459	0.4211
cat - dog	**0.9120**	**0.8556**
cat - tiger	0.8604	0.7492
cat - wolf	0.9017	0.5097

Table 5.1: Correlation of word co-occurrence frequencies

In SEMCOR corpus, the words *dog* and *cat* (in animal sense) appear 42 and 18 times, almost every instance referring to household pets; these two words come together in the same sentence 4 times, showing how much similarly the society sees them. The word *carnivore* never appears in the corpus.

A small experiment can be conducted to verify this issue further, systematically calculating word co-occurrence frequencies for words *dog*, *cat*, *carnivore*, *wolf*, and *tiger*.[1] **ANIMAL** senses of *wolf* and *tiger* share the same parent node with *dog* and *cat* respectively. Taxonomically speaking, *wolf* is much similar to *dog*, and *tiger* to *cat*. All instances of these five words were collected from the AQUAINT Corpus (Graff, 2002), and all words that co-occur with them, within a window of 50 words to both sides, were counted. The stop words were removed, and the rest of the words were compared for correlation in co-occurrence frequencies. The correlation values in co-occurrence frequency with other four words are shown in table 5.1 for 50 and 1000 words that co-occur most frequently with *cat*. It can be seen that *dog* shows the best correlation with *cat*. Figure 5.1 shows the co-occurrence frequencies of 50 words that co-occur with *cat* most frequently. Frequencies for each word is shown as a ratio to the co-occurrence frequency of the same word with *cat*; it can be seen that the plot for *dog* is the most parallel to that of *cat*. Both table and figure show that *dog* has best correlation with *cat* in terms of context words.

It should be noted that, some semantic similarity measures can partially handle this. For instance, in Jiang and Conrath (Jiang and Conrath, 1997) measure, (using SEMCOR corpus frequencies and the implementation of Patwardhan, Banerjee, and Pedersen

[1]Words are considered here, rather than senses, for the lack of a large sample of sense annotated data. As the relevant senses are the most likely ones for respective words (according to the WORDNET sense order), this does not pose a big problem, and can be hoped to work the way early experiments of Miller and Charles (1991) and Rubenstein and Goodenough (1965) did.

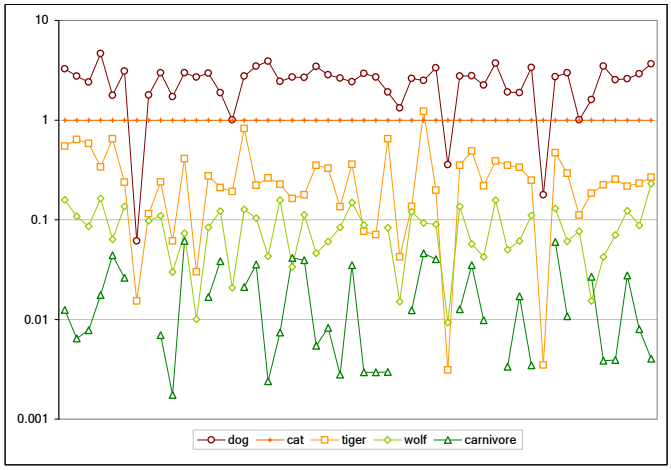

Figure 5.1: Word co-occurrence frequencies for *dog*, *cat*, *carnivore*, *tiger*, and *wolf*, for the 50 words which co-occur with *cat* most frequently (ignoring stop words). For easy comparison of co-occurrence frequencies of each word with *dog*, *cat*, *carnivore*, *tiger*, and *wolf* are shown as a ratio to the co-occurrence frequency of the same word with *cat*. Logarithmic scale is chosen to make ratio comparisons easy. Plot of *dog* is the most parallel with that of *cat*, showing best correlation in terms of co-occurrence patterns with other words in the context.

(2003)) the similarity between *cat–dog*, *cat–tiger*, and *dog–wolf* are respectively 0.5457, 0.2812 and 0.2852. However, this has nothing to do with real underlying semantics, as the higher similarity between *cat* and *dog* is just a result of the lower information content of the two words rather than their actual contextual similarity. For similar reasons, the similarity between *dog–carnivore* and *cat–carnivore* are 1.9576 and 0.7566 despite the much different correlation in usage. This is a structural problem in Jiang and Conrath measure that cannot be avoided; as shown in figure 5.2, concepts that lie on a sub section of a path between two concepts will always have a higher similarity value than the similarity value between the two 'terminal' concepts. By any means, Jiang and Conrath similarity measure does not take contextual similarity into account.

Wierzbicka (1996) provides many examples that seem to support our opinion on this. Despite the fact that her comments are not focused on WORDNET , they are readily applicable to our situation. One such issue is the problem of hypernymy that was mentioned above. For instance, take the two sentences

a. There is an insect on your collar.

b. *There is an animal on your collar.

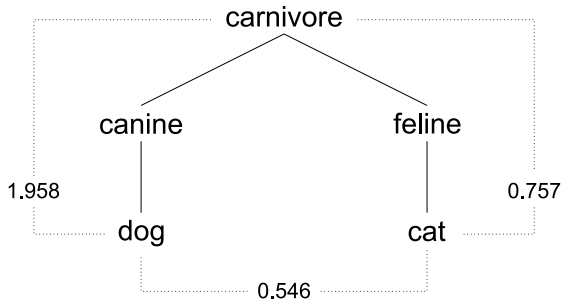

Figure 5.2: Jiang and Conrath similarities between *dog/1*, *cat/1*, and *carnivore/1*. Although better than most other similarity measures, JCn measure does not take contextual similarity in to account.

a. She is dancing joyfully.

b. *She is moving joyfully.

Although in terms of WORDNET hypernyms *insect* is a kind of *animal*, and *dancing* a manner of *moving*, a native English speaker would find the (b) sentences rather odd. Similarly, some words in English do not get used as much as the others, and make odd-sounding sentences.

a. Look at that animal/horse/cat.

b. *Look at that mammal/quadruped/carnivore.

5.1.1 Taxonomy vs. Usage of Synonyms

We need not go for hypernyms to find similar oddities; synonyms, for instance, behave in much different ways from each other. WORDNET has *wife* and *married woman* as synonyms, however one can hardly expect to find *married woman* in the same linguistic context as *wife*. Other similar examples are *car* and *machine*, *ant* and *pismire*, or in case of verbs *draw* and *delineate*.

This fact can post a problem to some approaches we discussed earlier in this section, regarding the use of substitute training examples. For instance, the first sense of *dog* (*a member of the genus Canis*) has two synonyms, *domestic dog* and *Canis familiaris*, both of which are monosemous; in theory, *domestic dog* and *Canis familiaris* should provide substitute training examples for *dog*. This is the same sense that involves dogs as

house pets; however, both synonyms are more inclined to be taxonomical and scientific, and it is very unlikely that both senses would be used in the same context *dog* is used in common English usage. For instance, we queried AQUAINT corpus (Graff, 2002) for the three items in the synset: First 100 instances out of 25079 instances[2] of *dog* all referred to either dogs as pets or police/guard dogs. Only three instances of *Canis familiaris* were found, all of which clearly referred to scientific sense. All 21 instances of *domestic dog* were either from scientific articles, or statistical references where dogs were contrasted with other non-domestic animals. Most of these articles also included words such as *experiment*, *study*, or *scientist*, showing the clear difference in the context. Table 5.2 shows a sample of five instances each for *dog* and *domestic dog* in AQUAINT . The differences in context is very clear; former is used in more 'domestic' contents, involving peoples day-to-day life issues such as urban life and pets, while the latter does not involve dogs' link to peoples domestic lives. This shows the problematic nature of using monosemous relatives of senses as substitute examples in particular, and using the taxonomy as a basis for sense classes in general.

The creators of WORDNET (Tengi, 1998, page 112) acknowledge this issue, and introduce a measure called *familiarity index* for words, which they suggest as a tool to differentiate 'layman' terms from scientific ones. The authors claim that the familiarity of a word is a crucial measure about that word. It can affect many factors of linguistic usage, such as the ease of reading, ease of understanding, the frequency of usage and the ease of recall when queried. They also claim that the frequency of usage is the best index of the familiarity of a word. But this measure, in the context of WORDNET, suffers from a problem: the amount of labeled data available from the SEMCOR semantic concordance is not large enough to reliably estimate frequencies.

For this reason, WORDNET uses an alternative measure of familiarity index, which is based on number of senses a word has, which will mean that *domestic dog* and *Canis familiaris* are less familiar than the word *dog*. This hints that the words that have many senses are more frequently used as well. Underlying assumption is that the number of

[2]when using the indexed retrieval system MANAGING GIGABYTES (Witten, Moffat, and Bell, 1999); in order to avoid bias, only one example from each resulting document was retrieved.

a. ...after a housewife was bitten by a **dog** last Wednesday ...
b. ...public appeal for control over dangerous or bigger breeds of **dogs** ...
c. ...while he was taking his **dog** out for a walk near his home ...
d. ...public housing estate tenants with **dogs** could be evicted ...
e. ...color television sets, ...rubies and cat and **dog** food ...

a. ...copying the familiar **domestic dog**, however, will force scientists ...
b. ...coyotes and **domestic dogs** result in multiple calls from prairie dogs.
c. ...do coyotes often mate with **domestic dogs** to produce what are called ...
d. ...wild dogs ...infected with ...none contact with the **domestic dogs**. .
e. ...experimenters ...breed of fox has evolved, ...look like a **domestic dog**.

Table 5.2: Instances of *dog* and *domestic dog* in AQUAINT corpus, showing why synonyms do not always provide good substitute training examples. Only one sentence from each document was extracted in order to avoid skewness. First five returned instances for each query shows the clear difference in contexts the two entries are used, although they are synonymous in WORDNET. The word *dog* is used in domestic/pet context and *domestic dog* in scientific context.

senses a word has is correlated with the frequency of word usage. This is a previously made observation by other researchers such as Jastrzembski and Stanners (1981; 1975) and Zipf (1945). However, this method is not foolproof either. Regularly used words such as *animal* and *knowledge* have only one meaning, thus WORDNET familiarity measure for them is 'very rare'.

These observations summarize the problem of inferring similarity from taxonomy. However much the WORDNET hierarchy based substitutions are taxonomically logical, we cannot expect them to work well as substitute examples, as long as they do not *behave similarly* in natural text, thus rendering their relationships useless in terms of textual features. They would have helped in a case where we used inference for WSD. The state of the art of WSD, however, does not make use of inference.

5.1.2 Taxonomy vs Semantics: Kinds and Applications

The reason for the problems observed by Wierzbicka is that the semantic classes of concepts can be much different from the taxonomical classifications. For instance, *cat* and *dog* fall into a semantic category of *pets*, which is a contextual classification rather

than scientific. Similarly, *pigs*, *cattle* and *lambs* can be described as *livestock* rather than *mammals*. The latter words will obviously share some contextual similarities, which are different from *pets* or from wild animals. Similarly, *birds* and *fish* will be seen differently from *animals*, as in saying *'we went to zoo and saw a lot of animals, birds and fish'*. (Compact Oxford Dictionary indeed lists as one sense of *animal*, *'a mammal, as opposed to a bird, reptile, fish, or insect'*. WORDNET entry for *animal*, interestingly, does not leave much room for interpretation. There is only one sense, which is *'a living organism characterized by voluntary movement'*.) Wierzbicka (1996; 1984) says that if there is a zoological 'unique beginner' in ordinary English, it is **CREATURE**, not **ANIMAL**.

This problem has effects within individual sense definitions at times, as the 'real world' sense can be much different from the 'taxonomical' sense. For instance, the gloss of sense 1 of *dog* is *'a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds'*. The same for *cat* is *'feline mammal usually having thick soft fur and being unable to roar'*. Both only annotate the concepts merely as specific kinds of animals, and forget the contextual semantics in commonsense, which would include the common domestic position of these two animals as pets, which make the language about them much different from that about rest of the carnivores. Wierzbicka (1984) notes that there can be different kind of hypernym relations. One is scientific taxonomy as is the case with WORDNET in general. Some others can be based on their utility value: for instance, to classify *fishing rod* as a type of *sports equipment* rather than a type of *rod*. In WORDNET, the hypernym of *fishing rod* is *rod – 'a long thin implement made of metal or wood'*, which does not relate to common usage of a fishing rod. Yet another method is to classify concepts according to the origin of them, as is the case with *leftovers* or *belongings*. Whether taxonomical classification is better than other ways of classifying senses into classes, for the end task of WSD, is not clear.

This problem is not a new issue, and WORDNET creators acknowledge the fact that "WORDNET itself does not give any information about the context in which the word forms and senses occur" (Fellbaum, 1998a, p.12). They note however, following Miller and Charles (1991), that word knowledge of a speaker involves not only the dictionary

meaning of words, but also the types of context in which they can appear.

5.2 Issues regarding WORDNET Structure

Some of the problems are related to the fact that WORDNET hierarchy itself is not designed keeping generic sense class based WSD in mind.

5.2.1 Hierarchy Issues

Our class system is meant for classes that are disjoint sets of senses. Although WORDNET lexicographer files satisfy this property at fine grained sense level (one sense being included in one and only one LF), when it comes to semantic structure, some LFs are subsets of others.

Figure 5.3 shows the structure of WORDNET noun hierarchy. Some LFs, such as **ANIMAL**, **PLANT** and **PERSON** fall under one super-concept **ORGANISM**. (WORDNET has a different lexicographer file **NOUN.TOPS** to accommodate the super concepts such as **ORGANISM**, which are not lexicographer files themselves). Some LFs fall under others, such as **FOOD**, which is a sub concept of **SUBSTANCE**, and **PROCESS** which falls under **NATURAL PHENOMENON**. This can possibly create confusion when learning, as there can be features common to both classes.

The lexicographer file **NOUN.TOPS** itself imposes a problem as it does not have a semantic meaning but a technical one. It contains all concepts that are marked as tops in the rest of the noun LFs, with 78 concepts and 11,917 instances. This is 5.2% of all labeled noun instances. One can overcome this problem by carefully assigning senses in the LF **NOUN.TOPS** into their 'rightful' LF owners — such as *animal* sense 1 to LF **ANIMAL**, but the assignment is not straightforward in many cases, such as *entity* and *organism*, which lie above the defined 25 LFs.

A similar situation is seen in 'stative' verbs as well. The verb lexicon is divided into two large groups which contain activities (such as *run*, *write*, *swim*) and events (such as *rain*, *burn*, *grow*) on one side, and states (such as *feel*, *rank* and *belong*) on the other. The former group consists of 14 lexicographer files, while the latter takes up the

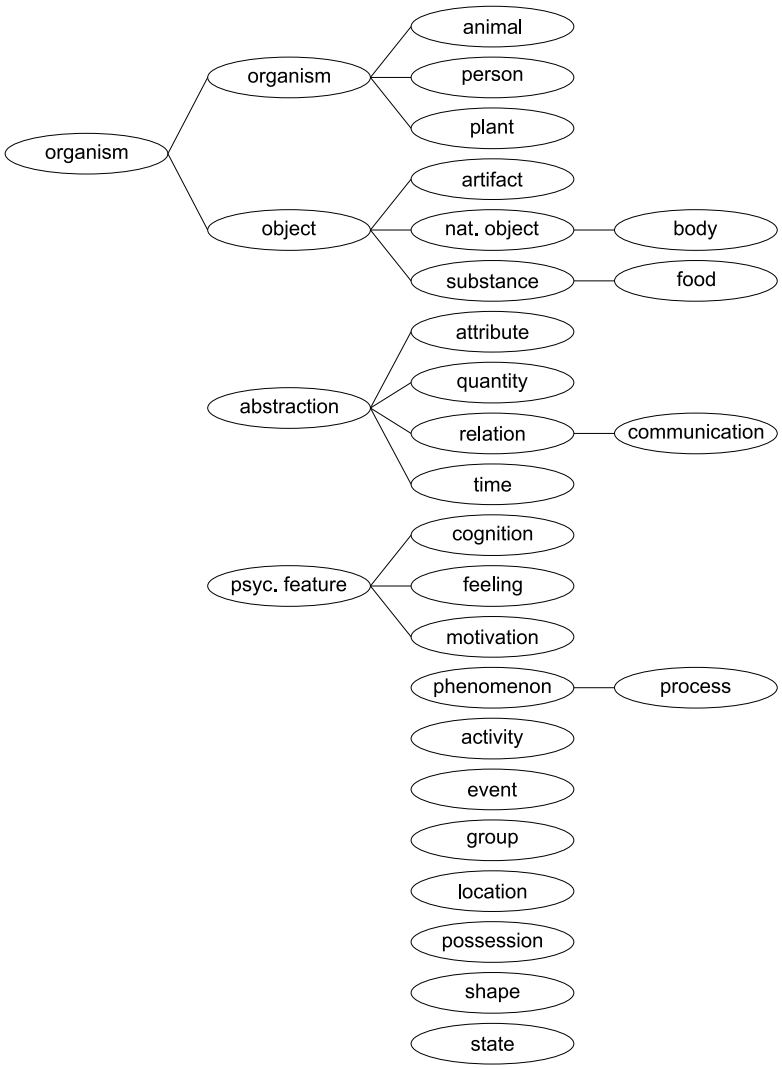

Figure 5.3: Organization of WORDNET noun hierarchy (Fellbaum, 1998a)

remaining one. According to WORDNET (Fellbaum, 1998b, p.70), this latter group is arguably made up of derivatives of *be*, which "constitute a semantically heterogenous class" and is the "only one group that does not constitute a semantic domain". In addition, it contains auxiliary and control verbs, (*will*, *shall*, *can*, *want*, *succeed*) and aspectual verbs (showing temporal aspect of an act or event) such as *begin*. The authors of WORDNET identify the issue that many verbs cannot be distinguished as belonging to one particular class, and admit that the differences are vague.

The inclusion of a large group of senses, especially very common ones, into one group makes the **STATIVE** group rather large. It occupies more than 20% of all instances of verbs, and in many cases, multiple senses from each word. The former factor is a problem in learning as the large number of instances add even spurious example instances that are statistically stronger than a word sense with only few instances. The inclusion of many senses of the same word in a large all-inclusive group makes the sense loss (see section 2.5) large for that word. Proportional sense loss is not exceptionally high in this class compared to other words; however, the large number of sense instances that are included within this particular class shows that if the sense loss for this class alone can be reasonably reduced, it will help increase the final fine-grained sense accuracy.

5.2.2 Sense Allocation Issues

The distribution of instances among LFs is not even; some like **ARTIFACT**, **ANIMAL**, **PLANT** and **PERSON** take up more than 10% each of all senses for nouns. Similarly, **CHANGE** and **MOTION** verbs each take large portions of verb senses. The portions occupied by each LF are shown in figures 5.4 and 5.5. Shown as 'count' is the unique count of entries in WORDNET sense index for each lexicographer file, while 'total' shows the proportions each lexicographer files occupies in SEMCOR corpus. The latter gives a more realistic idea of the LF distribution in natural text.

When looking at actual sense allocation, it can be seen that some of these senses can better be split into more cohesive and meaningful clusters. Taking the familiar **ANIMAL** LF as an example, as we discussed in the pervious section, it is possible to

have classes like **MAMMAL**, **BIRD**, and **FISH** which will be more cohesive in terms of features. Similarly **ARTIFACT** can be split into **BUILDING**, **VEHICLE**, and so on, and verb class **CHANGE**, into **FORMATION**, **INCREASE**, **SEPARATION** and so on.

If we consider the actual instances, **ACT**, **COMMUNICATION** and **GROUP** for nouns and **COMMUNICATION**, **SOCIAL** and **STATIVE** for verbs look as better candidates for split.

On the other hand, LFs such as **COGNITION**, **FEELING**, and **MOTIVE** might not be separable from each other using contextual features alone. They could possibly be merged into their common parent, **PSYCHOLOGICAL FEATURE**, without losing much information. Even in the sense map, these three take only 4% of senses and 8% of all instance together.

In fact, as we mentioned earlier, even single senses sometimes look too broad in definition. The word *animal*, for instance, has only one sense *'a living organism characterized by voluntary movement'* and *knowledge*, only sense *'the psychological result of perception and learning and reasoning'*. The first definition of course forgets the derogatory use of *animal* for humans and for any human types, such as in *a political animal*. The American Heritage Dictionary of the English Language (Pickett and others, 2000) lists 5 noun senses for the word. The same dictionary provides 6 senses for noun *knowledge*.[3]

There are also instances where the lexical file associated with a sense is not always obvious. For instance, *ionosphere/1* and *stratosphere/1* senses are assigned different lexicographer files, namely **LOCATION** and **OBJECT**. This kind of assignments can make it difficult to automatically learn the classes from the context, as the underlying concepts have contexts very similar to each other, but fall in to different classes.

5.2.3 Large Sense Loss

Recall from section 2.5 that we defined sense loss as the proportion of senses that we cannot handle due to the constraint that only one fine grained sense can be mapped from a sense class. Although it was argued that this loss is reasonably smaller for

[3]However there is no way to handle this issue within the generic sense class framework as it does not support splitting fine grained senses.

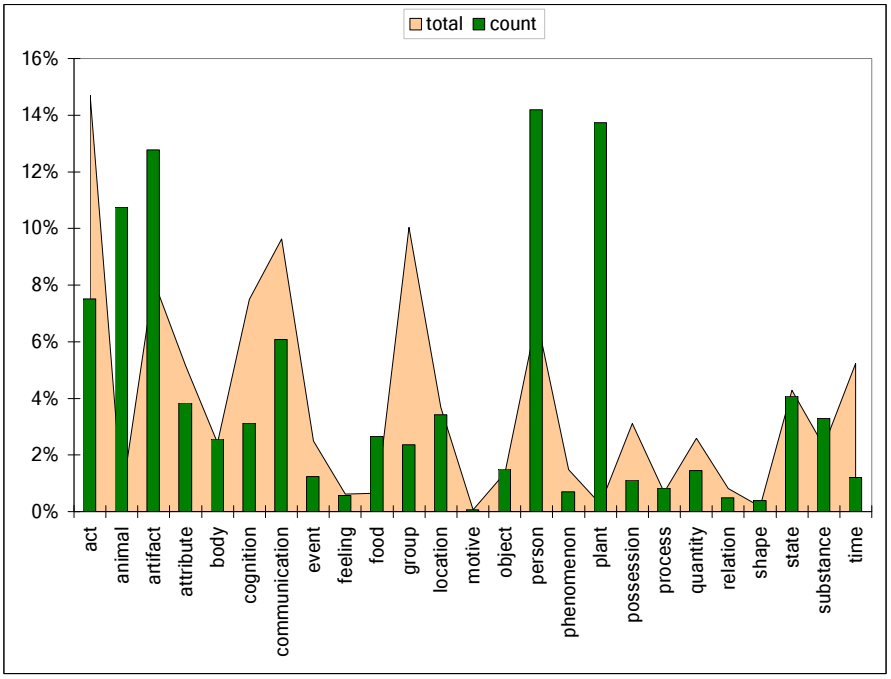

Figure 5.4: Proportions of number of senses each lexicographer file occupies in noun senses. Bars show the counts of unique senses in WORDNET, while the areas show the total number of instances occupied by each lexicographer file in SEMCOR corpus.

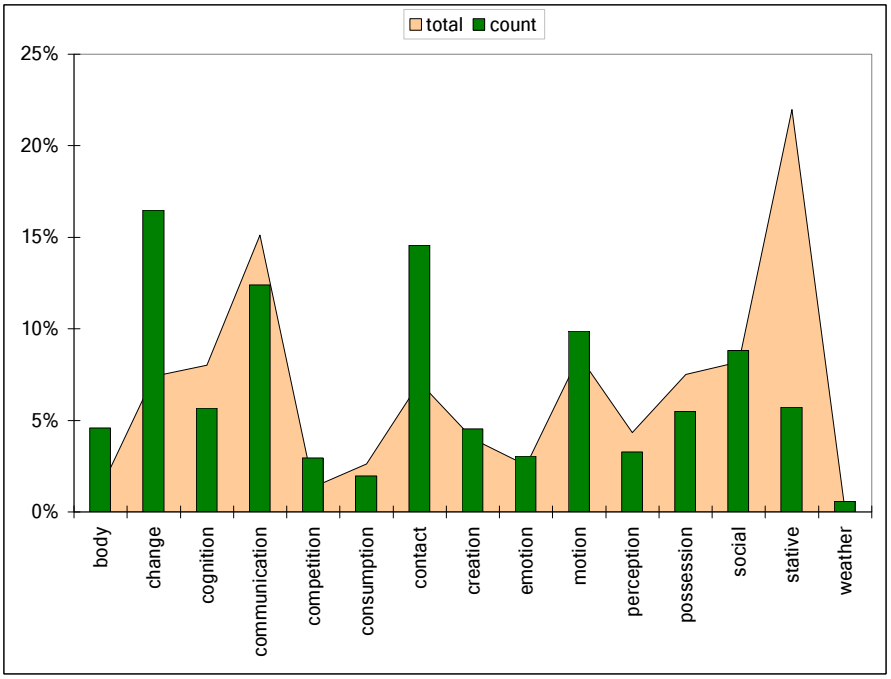

Figure 5.5: Proportions of number of senses each lexicographer file occupies in verb senses. Description as per figure 5.4.

WORDNET lexicographer files, the fact remains that the number of semantic classes the LFs provide is fairly small; from one side, this can exacerbate the problem of feature incoherence as there are smaller number of semantic 'bins' to put every sense we find; on the other hand, even when it is possible to accurately classify previously unseen instances into their correct classes, the system would still have a large sense loss, as there is a good chance of several senses of the same word falling into the same class. This is specially critical for verbs, as verbs generally have more senses per word and are divided to only 15 lexicographer files. As we showed in figure 1.2, 23% of the instances of polysemous senses in SEMCOR corpus are lost for verbs, as compared to 14% for nouns.

If we consider the loss on the instances that are *correctly classified* at the class level, instead of considering the *theoretical* sense loss, this pattern is even more prominent for verbs. Tables 4.17 shows that the out of the instances that were labeled with the correct class, only 63% made it to the correct fine grained sense in SENSEVAL-2 and only 70% were correctly covered at fine-grained level in SENSEVAL-3.

5.2.4 Adjectives and Adverbs

The lexicographer files do not have a semantically organized structure for adjectives and adverbs. Adjectives have only three lexicographer files, while adverbs have only one. The classification is not based on even a taxonomical semantics. The framework presented in the previous chapters could not be implemented with existing WORDNET lexicographer files for these two parts of speech.

If one can derive a set of classes with enough resolution, it would be possible to use the same framework for these two parts of speech as well.

5.3 Classes Based on Contextual Feature Patterns

One approach that can be used to address the issues discussed above is to base our set of classes on the features found within the text itself. This kind of classes will be consistent in terms of features, and will pose no problem about usage patterns, as the

classes are based on the similarities and differences of same usage patterns that the classifiers will later use to learn them.

Lexicographers have always been using frequently occurring contextual patterns to identify particular senses of words. Even some current work such as WORD SKETCH (Kilgarriff and Tugwell, 2001b) and WASPS (Tugwell and Kilgarriff, 2000; Kilgarriff and Tugwell, 2001a) are good evidence how the patterns among syntax can lead to discovery of senses, rather than starting from a taxonomy of senses and then identifying the patterns that represent each sense.

There has been several research work in and outside computational linguistics, which focused on the syntactic features and relations, rather than purely taxonomical notions, to classify words into groups. Some of these were discussed in sections 2.6.2 and 2.6.3; a few deserve some attention here.

The work of Levin (1993) discusses in great detail how words can be classified according to so-called *diathesis alterations*, syntactic constructs based on the allowed alterations of expressions and arguments. She discusses how a speaker of a language can infer acceptable forms of usage for a verb from its meaning, and argues that "the picture that emerges is that a verb's behavior arises from the interaction of its meaning and general principles of grammar" (Levin, 1993, p.11) and also that "the ties between a verb's meaning and its syntactic behavior cannot simply be ignored" (p. 12).

According to Levin, the study of diathesis alterations is important, as it is difficult to classify verbs using manual observations alone. The patterns found using the similarities/dissimilarities of diathesis alterations can be used as insights into meaning, thus facilitating the classification of verbs.

Levins work provided some inspiration on our idea of classifying words purely on syntactic and lexical patterns. For her work it was necessary to understand the semantics of the classes once they are formed. For the end task we have at hand, the need for figuring out the underlying *semantics* of the classes does not even arise; all we need is an acceptable and coherent level of similarities in sense behavior within a class, so we can learn them using an automatic classifier.

5.4 Summary

WORDNET LFs are an intuitive first choice for generic word sense classes. However, in a machine-learning based WSD perspective, they happen to be associated with considerably problematic issues. The fundamental reason for this is that the LFs are not meant to be learned from a labeled corpus using the contextual features we can find from within text, but are based on the underlying taxonomical semantics that humans understand. However semantically sound the lexicographer files are, they are admittedly not designed keeping WSD in mind; the problems discussed in this chapter suggest that it may be possible to improve the set of sense classes in such a way that they better suit the fine grained WSD task at hand.

Any attempt for remedying this needs to keep in mind the two design objectives for the new class set: coherence in terms of textual features, and smaller sense loss.

Whenever you find that you are on the side of the majority,
it is time to reform.
— Mark Twain

Chapter 6

Sense Classes Based on Corpus Behavior

In the previous chapter it was noted that, although WORDNET lexicographer files can be used as generic classes, there is no confirmed evidence that they make an optimal set of classes. WORDNET lexicographer files were instrumental in demonstrating the practical feasibility of learning generic word sense classes for WSD. However, as we discussed in the previous chapter, they come with significant problems attached. Some of the issues are related to the practical machine learning problem, which was not an objective in designing WORDNET lexicographer files. Some others are related to the end task of fine grained WSD. This observation points us to the main contribution of this thesis.

Recall from sections 2.4 and 2.5 that the only requirement for a generic class system in our framework is that it should provide a many-to-one mapping from fine-grained senses into classes, and there must be an ordering of preference for senses, which is used in the reverse mapping. These conditions are easy to satisfy, and leave some degree of freedom for any kind of possible optimizing.

Consider a set of classes that is directly based on *linguistic usage patterns*, rather than a semantic taxonomy. It intuitively follows that such a set of classes would be easier to learn using a set of features that reflect the same usage patterns. Note that there need be no linguistic assumption made here about the classes. The only necessary assumption

is that the features used in the process of identifying the different classes will show similar properties in the classification phase as well, with a different set of instances. This is not very far from the fundamental hypothesis of inductive learning (Mitchell, 1997).

This formulation does not explicitly depend on taxonomical semantics; as long as the usage patterns of senses form cohesive classes, our WSD framework is supposed to work on them.

Applying some clustering technique can be suggested as a means to find a system of classes that are cohesive *in terms of features*. In this chapter, such a system is described in detail, with a discussion of technical and practical issues; some properties of the classes so-formed are also analyzed along with their performance in fine grained WSD.

6.1 Basic Idea of Clustering

Justification for contextual behavior based sense classes came from the previous chapter. What is needed to be done is to generate a set of classes that are consistent in terms of the same features that are used in the WSD classifier task. This type of problems can be solved by computational approaches, using clustering techniques.

Clustering scheme proposed here is based on features of text, namely local context and parts of speech. These features are used in a clustering algorithm in order to build classes of senses that are *behaviorally* similar to each other: Senses are represented by their features in labeled corpus instances; a clustering algorithm can find out the instances that have similar features (which means they have similar corpus behavior) and puts them in the same cluster. A cluster obtained this way is essentially a sense class, although one might not be able to identify any meaningful semantic label for it.

Clustering senses using linguistic features is not a totally new idea: one can find a relationship of this idea with thoughts of early philosophers. As early as in 1918 Wittgenstein argued that senses are defined by their uses, as opposed to the Aristotelian idea of referential interpretation of word meaning; Antonine Meillet argued that the sense of a word is determined by the average of its linguistic uses (Ide and

Véronis, 1998). Lexicographers have been using similar techniques to identify new senses, by analyzing frequent usage patterns in large corpora of text (Tugwell and Kilgarriff, 2000). Wilks (1997, p. 81) also mentions about a 'dictionary that consisted entirely of usages' although saying that the use of such a dictionary is unclear. Some unsupervised systems use similar ideas of clustering similar contexts for practical WSD, among other possible uses (Pedersen and Kulkarni, 2005). In a way, the proposal for finding substitute senses using their lexical and syntactic features alone could be interpreted as a possible use for a 'usage based dictionary' or rather a thesaurus based on syntax rather than semantics.

6.2 Clustering Framework

Most clustering algorithms work on coordinates, and it is necessary to represent word senses as coordinates. This is done by converting local context and parts of speech vectors (formed in the way described in section 3.1.3) into binary vectors. The technique of creating the binary vectors was the same as described in section 3.4.1. Also, since parts of speech and collocation provide two different types of information, only one type of features was used at a time. Without this, statistically strong part of speech features dominate the feature vector when building the coordinate vectors by dimension reduction (see section 6.2.1). The singular value decomposition technique used for dimension reduction ignores the classes of instances, and does not consider which features give the best information. The much small number of possible feature values for the parts of speech feature compared to local context (about 30 per window position, compared to about 7000 in local context) makes them statistically strong, and dominate the coordinate vectors. This is undesirable, as it was seen (for instance, in section 4.3) that the local context features are the most important. In addition, separately working on the two types of features enables comparison of two features independently. Syntactic pattern feature was not used because it was much sparse compared to other two types.

The first problematic issue with regards to sense 'coordinates' stems from the fact

that there can be multiple labeled instances for a sense in the corpus; taken as individual instances, different instances of the same sense do not necessarily clump together in the instance space. Instances of different senses are distributed in such a way that areas occupied by individual senses overlap with each other. This is mostly due to outliers of senses. This is not surprising; there is no obvious contrast in contexts of different senses, as amply shown by the poor performance of WSD systems in general.

This is a problematic issue, as we have to satisfy at least the minimal constraint that different instances of the same word sense fall into the same cluster. Without satisfying this condition, resulting clusters will be useless for any purpose.

One intuitive way to attain this objective is to introduce a system of hard constraints that the instances of the same sense must be linked to each other. Some clustering algorithms, constrained k-means algorithm (Wagstaff et al., 2001) for instance, provide a method to achieve this. Another possible way to run an agglomerative clustering algorithm with a starting state where different instances of the same sense are already agglomerated together. Unfortunately, the conditions in our case are extremely unfavorable for this kind of techniques, as the must-link constraints link instances that are almost randomly distributed over the instance base. This causes clusters to form with very poor quality, without much coherence within clusters or much differences between clusters. As a result, such clusters will be impossible to learn from the features.

For this reason, a work-around solution based on smoothing had to be sought, rather than relying on hard constraints that link instances of the same sense. This was to average the coordinate vectors of all instances of the same sense, and use the centroid as the 'representative' coordinate of that sense. This allowed for some variation within sense, while making sure that individual spurious instances cannot add too much noise to the clustering process.

6.2.1 Dimension Reduction

The 'coordinates' obtained this way have high dimension, especially in the case of local context. The use of Singular Value Decomposition (Golub and Reinsch, 1970; Wall,

Rechtsteiner, and Rocha, 2003) is a common technique in cases like this for reducing dimension of data.

Let us assume that there are m number of features in the vector, and we have n number of senses; the sense-feature representation in this case would be a matrix X of dimensions $m \times n$, where each column will denote a given sense. With singular value decomposition, we decompose X into three matrices

$$X = USV^T$$

where U is $m \times d$, S is $d \times d$, V is $n \times d$, and $d = \min(m, n)$. S is a diagonal matrix; values in its diagonal are called *singular values*. U and V have orthonormal columns, and are respectively called left- and right- singular vectors.

We assume that the singular values $s_1, s_2, \ldots, s_d$ of S and columns of U and V are ordered in such a way that $s_1 > s_2 > \ldots > s_d$; then, we can keep only $k(< d)$ largest singular values and set the rest to zero; this is effectively similar to reducing the dimension of rows of U and V from d to k. The result of their multiplication, $\hat{X} = V_k S_k U_k$, will provide the closest k rank approximation to X, in the sense that the sum of squared distance between the elements $\sum_{i,j} |x_{i,j} - \hat{x}_{i,j}|^2$ is minimized.

In our case, we use the scaled senses $S_k V_k$ with the reduced dimension as representative coordinates of the senses, and discard the feature vectors in matrix U. The magnitude of singular values has a sharp drop, and all components with singular values smaller than 1% of the largest one were discarded. The resulting coordinates are normalized to avoid any further effect from the number of samples per sense. Averaging before SVD does not totally handle this, as the resulting vectors after averaging have different Euclidian lengths, and SVD process generates vectors of different magnitudes than the originals.

Coordinates obtained this way are used in two different clustering schemes, which are described in sections 6.3 and 6.4.

6.2.2 Standard Clustering Algorithms

Before resorting to the clustering algorithm described in what follows, experiments were conducted to test several standard clustering algorithms. The best algorithm was chosen based on the fine-grained WSD performance of the resulting sense classes on the development data set. This is a justifiable approach as there is no theoretical way to predict how well the sense classes would perform on fine-grained WSD, which is our sole aim. On this performance measure, the clustering algorithm described in the next section fared best. A Description of the implementation and evaluation of other clustering algorithms is given in Appendix A.

6.3 Extending k Nearest Neighbor for Clustering

A modification to k-means algorithm, K-means+ (Guan, Ghorbani, and Belacel, 2004), provided the basis for our clustering. The algorithm dynamically determines the number of clusters depending on the variance properties of the instance base. The algorithm works as follows:

Clustering is initialized with the same number of clusters as the number of lexicographer files for that particular part of speech. Each sense is assigned to the cluster corresponding to its lexicographer file. After this, the clustering proceeds a way similar to k-means clustering.

6.3.1 Algorithm

1: Set the number of clusters to the number of lexicographer files.
2: Initialize the clusters by assigning each sense coordinate to the corresponding cluster, determined by its lexicographer file.
3: calculate the centroid of each cluster, and reassign the coordinates to the cluster whose centroid is nearest to it.
4: calculate the variance of each cluster. Variance V_i for cluster C_i with centroid c_i is defined as $V_i = \sum_{s_j \in C_i} |s_j - c_i|^2$.
5: let threshold variance $V_0 = \max_i(V_i) - \delta$, where δ is a small constant.

6: **while** not converged **do**
7: **for all** cluster C_i **do**
8: **for all** sense S_i such that $s_j \in C_i$ **do**
9: **if** $|s_j - c_i|^2 > V_0$ **then**
10: create a new cluster with s_j as its sole member.
11: Add the new cluster to the set of clusters.
12: **end if**
13: **end for**
14: **end for**
15: calculate new centroids of clusters.
16: reassign all senses according to new centroids.
17: **for all** cluster C_i **do**
18: **if** number of members in $C_i < N$ (where N is a constant) **then**
19: $parent(C_i) = \arg\min_j(distance(C_i, C_j)), j \neq i$ where
$distance(C_i, C_j) = \min_{k,l} |s_k - s_l|^2 \; \forall s_k \in C_i$ and $s_l \in C_j$.
20: merge C_i with $parent(C_i)$.
21: **end if**
22: **end for**
23: **end while**

Steps 17 – 22 ensure that all clusters are above a certain size, but still avoids large clusters by allowing the clusters to be non-spherical.

Constant δ was arbitrarily set to 0.01 as the only need for it is to provide minimum condition such that the 'seeding' starts. The number of minimum members per cluster, N, is a determining factor of the number of clusters, and was selected by experimenting with a range of numbers; cross validating on the held out data set, and picking the number that yielded the best results.

Parts of speech and local context features led to two different clusterings, both of which could locally optimize the quality of clusters in terms of sum of squared distance from the points to the cluster centroids.

In addition to nouns and verbs, adjectives could also be used with this clustering algorithm, as the algorithm does not depend on a hierarchy. In case of adjectives, clusters were initialized with the three lexicographer files available (although they do not have any semantic use), and were allowed to split. Adverbs however were left out due to practical reasons: One reason is that they have only a small number of words (only 269) with polysemous senses that have instances in SEMCOR corpus. Without any labeled instances, determining which cluster a sense belongs to is impossible. For this reason, and because of other constraints on resources, it was decided to leave adverbs out.

6.3.2 The Direct Effect of Clustering

Figures 6.1 and 6.2 show the direct effect of clustering on the instance base. The Euclidian distance between two coordinates c_1 and c_2 is equal to $|c_1 \cdot c_1 - 2\, c_1 \cdot c_2 + c_2 \cdot c_2|^{1/2}$. Since our coordinates are normalized, they have unit lengths, and this simplifies to $|2(1 - c_1 \cdot c_2)|^{1/2}$.

Given a d-dimensional coordinate system for n number of senses as an $n \times d$ matrix C, we can calculate the inter-sense euclidian distance matrix $D_{n\times n}$ by

$$D_{n\times n} = \sqrt{2([1]_{n\times n} - C \cdot C^T)}$$

The distance between the coordinates of sense x and sense y is represented by the pixel at (x, y) and also at (y, x). As the coordinates are normalized they all have the same length of 1, so the distance between two 'senses' can vary from 0 to 2. The matrix shown in the images show inter-sense similarity, in a color-coded scale with black showing zero distances, and white showing distances of 2, and gray shades showing respective values in between the scale. Several points are evident:

- Local context features have typically weak coordinate values, showing the larger differences between senses due to higher dimension. Parts of speech based coordinates allow instances to be more similar, showing larger content of black.

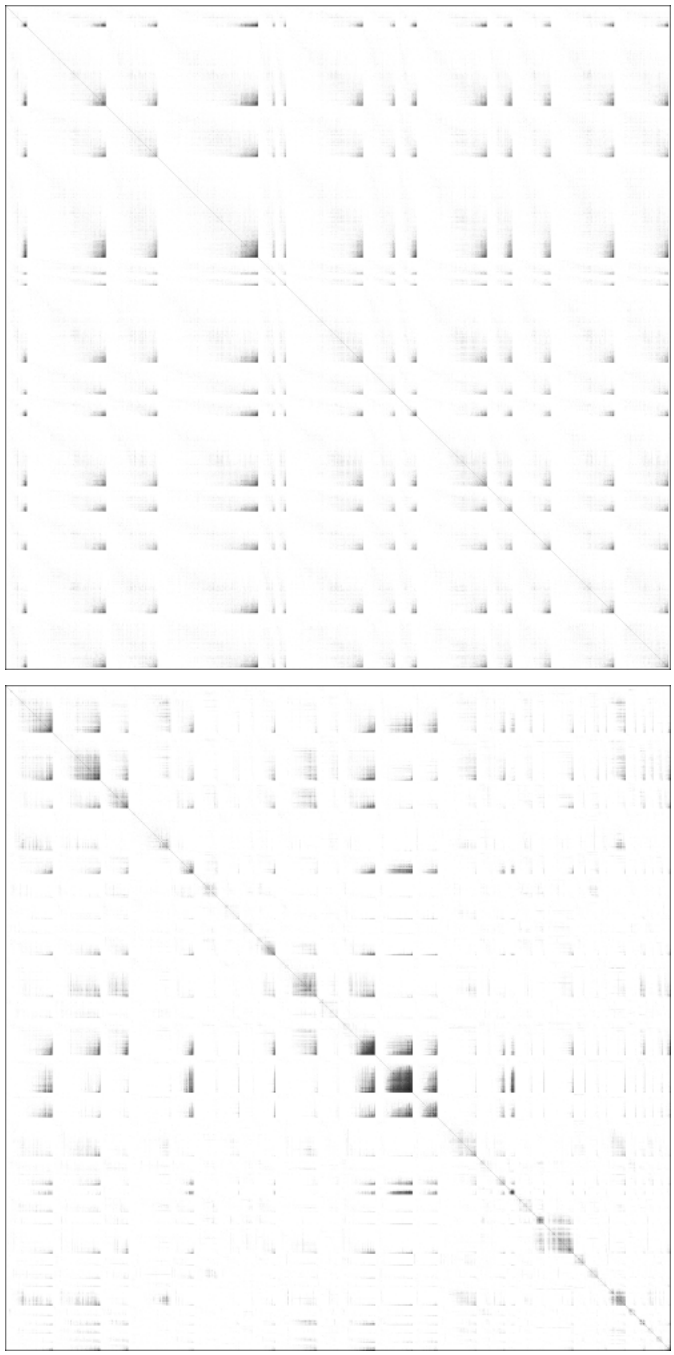

Figure 6.1: Proximity matrix in verb clusters for local context features. Inter-centroid Euclidean distances for senses, using coordinates after singular value decomposition. Above: original WORDNET LFs, below: feature based clusters. Darker colors show smaller distances. Emphasis on diagonal component of feature based based clusters shows the increased cohesion of features within the cluster.

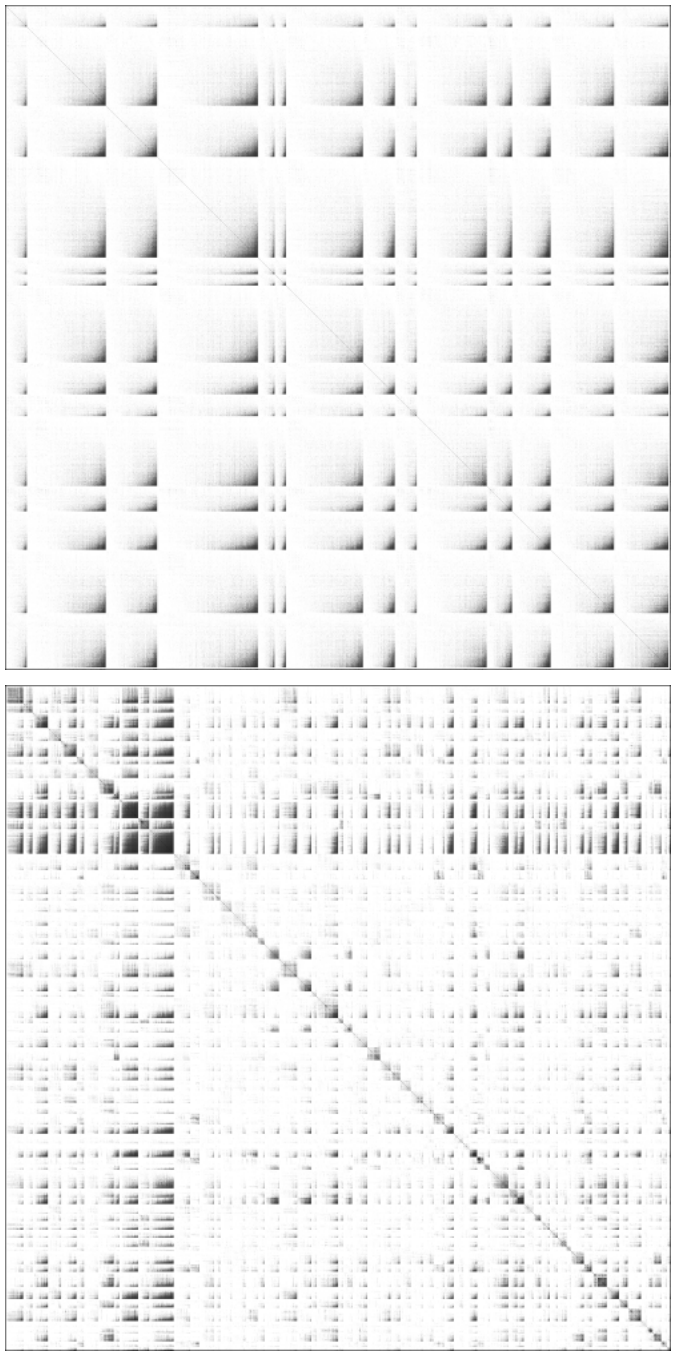

Figure 6.2: Proximity matrix in verb clusters for local context features - description as per figure 6.1. Above: original WORDNET LFs, below: feature based clusters. The differences are clearer than for local context because the feature space is much dence, due to the limited number of possible POS tags available.

- In the original lexicographer file clustering, the inter-cluster similarity patterns are not much different from that of intra-cluster similarity, as shown from the lack of a prominent diagonal. In other words, all classes are more or less similar to each other in terms of feature coordinates.
- Both feature based clustering schemes show a prominent diagonal, although weak in local context, showing that the intra-cluster similarity has been improved by clustering.
- In all cases, some clusters seem to have more 'generic' coordinates that make these clusters considerably similar to all other clusters.

A qualitative analysis on the end task will be provided in sections 6.7.2 and 6.7.4.

6.4 Control Experiment: Clusters Constrained Within WordNet Hierarchy

One may argue that the negative issues regarding WordNet LFs, described in the previous chapter, are mostly due to the mere size of the LFs, and just segmenting WordNet lexicographer files into smaller, more cohesive clusters suffices to solve the problem. For instance, **ANIMAL** can be split into sub segments such as **MAMMAL**, **BIRD**, **FISH**, **INSECT**, and so on. Miller (1998) acknowledges the possibility for the existence of such basic categories, and mentions that WordNet does not attempt to identify these 'basic level' categories, which he says lie somewhere in the middle of the hierarchy. For instance, the word *chair* is a well-defined concept, while its parent *furniture* has only the most general information, not that informative in visualizing it as an entity. The child concepts of chair, such as *throne*, differ from *chair* only in minor details. The concept *chair* in this case is a basic one.

This argument is still concerned on semantics only; it does not directly address the issues related to lexical or syntactic features we are concerned about, which we discussed earlier (in section 5.1). Still, it is worthwhile to examine how (and if) the cluster *size* alone can affect the quality of clusters. For this reason, a control experiment was

conducted to split WORDNET semantic hierarchy into smaller segments, keeping the structure intact – that is, senses will not change their relative positions in the hierarchy with respect to each other, but the branches of the hierarchy will be separated from the parent if they significantly differ from the rest. We can hope that this kind of segmentation will possibly be able to identify the some sort of 'basic concepts' similar to those mentioned by Miller, based on distinguishable syntactic/lexical properties of the segments within the hierarchy, rather than a manually made taxonomy.

The coordinates used in this experiment were the same as those used in the previous experiment (described in section 6.2.1), obtained from parts of speech and local context vectors after singular value decomposition.

6.4.1 Algorithm

A tree structure is built replicating the WORDNET hierarchy. the tree is populated with the coordinates, so that nodes are assigned the coordinated of their respective senses.

The tree can be segmented at any given node below the root of the hierarchy. Define *centroid* c_i of a given tree segment T_i as the average of coordinates of all senses that belong to the segment, and *variance* of the segment as the average squared distance of all senses within the segment, to its centroid.

At each step, each non-root synset (node in the tree) is considered as a candidate for splitting. A split causes the centroid to move within the remaining segment and the new section to have a new centroid, and effectively reduces the total system variance. The node that can yield the best reduction in variance is considered as the split point. The algorithm proceeds greedily by selecting at each step a node for split, until the required number of segments ('clusters') is reached. Similar to the previous method, the hierarchy is split into a larger number of segments than necessary, and smallest segments —typically consisting of single node outliers— are merged back into the tree. This is to ensure that the process does not yield undesirably small segments. However, in order to preserve the structural consistency, the small segments are merged back to the parent nodes from which they were broken away, rather than the nearest cluster, as in the previous case.

According to the 'official' WORDNET depiction of the noun hierarchy there are 11 'pure' unique beginners that do not have hypernyms (ref. figure 5.3); however, an analysis of nouns show that there are only 9 unique synsets that have no hypernyms: namely *abstraction/6*, *act/2*, *entity/1*, *event/1*, *group/1*, *phenomenon/1*, *possession/2*, and *psychological-feature/1*, *state/4*. Absent are *shape/2* and *location/1*. These 9 synsets encompass a total of 13 senses. Verbs have 626 parent-less synsets, which includes 1456 senses.

Just taking the second level of the hierarchy below the lexicographer file level, without programmatically splitting the hierarchy, could have been suggested as an easier way to produce a more refined level of abstraction: however this approach has at least two types of practical problems. First, some concepts that are not semantically significant lie as the direct children of lexicographer-file level unique beginners: for instance, *moon/2*, defined as *'any object resembling a moon'* is a direct descendent of **OBJECT**; *omniscience/1* is a direct descendent of **STATE**, and *might-have-been/1* (defined as *'an event that could have occurred but never did'*) descends directly from **EVENT**. This kind of children are clearly useless as generic semantic representatives. On the other hand, the top level nine noun synsets without parents, which were mentioned above, have 250 direct descendants. In case of verbs, this number is as large as 8152. This is too large a number of concepts than we would have liked for a set of generic sense classes, as an increase in the number of classes limits the number of available examples for a single class, and generally work against the aim of generalization.

Note that this clustering method requires a hierarchy, hence it is applicable only for nouns and verbs. Adjectives do not have a proper hierarchy although they are categorized into three lexicographer files. Hence we could not apply this method for adjectives.

6.5 Adjective Similarity Measure

Most similarity measures that are described in the previous chapters can be used only in a context where a hierarchy is present, which links the compared concepts (word

senses) together. This is the largely the case for nouns and verbs. (Although verbs do not have a single common 'universal' parent node like nouns do, verb senses can be compared at least where a common parent exists.) The lack of hypernyms for adjectives and adverbs makes it impossible for hierarchy-dependent similarity measures to be defined on them. The measures that do not depend on a hierarchy, such as the Lesk measure (Lesk, 1986), is usable in the case of adjectives. Unfortunately, as was the case with nouns and verbs (we discussed this in more detail in the section 4.5), the Lesk measure did not yield a reasonable performance.

Some previously published work employed an approach for capturing context information under sparse data conditions, using a method similar in theory to our idea of capturing 'similarity' through the local context vectors. This work, reported by Strapparava, Gliozzo, and Giuliano (2004) employs a classification scheme called WORDNET domains. The scheme, like that of Lesk, depends on textual context of the two concepts compared, and uses contextual coherence as a measure of similarity. They experimentally used a comparison scheme which used the Latent Semantics (Deerwester et al., 1990) of the term vectors instead of 'raw' vectors, and showed that this can outperform traditional context vector based sense comparison. Vectors in Latent semantic space were used to compare the context of the target word sense instance with the available 'document' vectors (built from known examples) of the different senses of the word.

This method could be adopted as a similarity measure; instead of directly classifying senses according to vector similarity, one can use the scaled vector dot product as a measure of sense similarity; this is essentially the same setting as Lesk measure, but when the vectors are converted to LSA space much of the noise is hopefully removed, as Strapparava, Gliozzo, and Giuliano (2004) suggested. The technique used in LSA based comparison is closely related to the singular value decomposition we discussed earlier in this chapter (in section 6.2.1). Remember feature vector matrix $X_{m \times n}$ of n sense vectors of m dimension was decomposed into

$$X = USV^T$$

where $S_{k\times k}$ is the diagonal matrix of principal components, and $V_{n\times k}$ the corresponding 'document', or in this case, sense vectors. The similarity between the senses are given by

$$Sim = V_k S_k S_k V_k^T,$$

where Sim is an $n \times n$ matrix, whose $(i, j)^{\text{th}}$ element represents the LSA-space vector product similarity between sense i and sense j. V_k and S_k are the reduced-dimension versions of V and S.

To avoid making issues complex by recalculating the LSA vectors for senses, a work around was employed; the principal components of the sense vectors have been already calculated to be used in clustering. Although most LSA systems would use a wider context than what was used for clustering, the nature of the LSA vectors are essentially the same in both cases. Therefore, it is possible to use the same principal component vectors output from the SVD in the clustering task. Interestingly, this approach seemed to work on development data set, and could achieve a small performance gain over WORDNET first sense for adjectives. This confirms the previous observations of Strapparava, Gliozzo, and Giuliano (2004), and hints that LSA can have wider scopes of applications in WSD.

6.6 Classifier

Once the classes are formulated, they can be evaluated on the fine grained WSD task, on the same framework as described in section 3.1. The only input fed to the WSD process from here is the class mapping, which is used the same way the WORDNET LFs were used.

Note that there are two different sets of classes, one purely based on features (section 6.3) and one constrained to WORDNET hierarchy (section 6.4). They each have two class maps, which are based on either parts of speech or local context features. All four class mappings are compared for performance in what follows.

6.7 Empirical Evaluation

In this section, the quality of these clusterings will be analyzed in terms of various measures that are relevant to the end task, as well as to the classification performance. We will also discuss the results for automatically created generic sense classes on SENSEVAL English all words task evaluation data.

The focus in this section will be on the clustering based on the algorithm that was discussed earlier in this chapter (section 6.3). As mentioned earlier, the relevent clustering algorithm was selected over other known clustering algorithms on the basis of development data set performance. Discussion of several representative standard clustering schemes, which were used in the experiment, but were discarded due to poorer development data set performance, is moved to Appendix A, for the sake of clarity and conciseness.

On a glance at the surface level, a casual observation of the finer grained class set, compared with a coarser set such as WORDNET lexicographer files, reveal many cases where obviously different senses that fall into the same WORDNET lexicographer file due to the coarse granularity of the LFs now fall into different classes in feature based clustering. For instance, the verb senses *address/2 (give a speech to)* and *address/3 (put an address on an envelope)* fall into different classes in both POS and local context feature based clusters, but they both fall into same lexicographer file **COMMUNICATION** in WORDNET . Similar behavior can be seen in nouns as well; for example *crusade/1 (a series of actions advancing a principle towards a goal)* and *crusade/2 (military expeditions in the 11-13th centuries)*, or *mate/1 (officer rank in ships)* and *mate/3 (partner of an animal)*, fall into different clusters in the new clustering, but into the same WORDNET lexicographer files.

Apart from obvious qualitative differences like these, it is possible to asses the differences that can be quantitatively measured.

System	Recall
Baseline (WORDNET first sense)	0.658
WORDNET LFs: k-NN	0.674
WORDNET LFs: SVM	0.670
Feature based clusters: k-NN	**0.687**
Feature based clusters: SVM	**0.682**

Table 6.1: Final results of feature based clusters on SENSEVAL-2 data. For comparison, baseline and previous best results using WORDNET lexicographer files are also given.

System	Recall
Baseline (WORDNET first sense)	0.643
WORDNET LFs: k-NN	0.661
WORDNET LFs: SVM	0.659
Feature based clusters: k-NN	**0.677**
Feature based clusters: SVM	**0.667**

Table 6.2: Final results of feature based clusters on SENSEVAL-3 data.

6.7.1 Senseval Final Results

Before analyzing the results in detail, a final summary results for SENSEVAL tasks will be provided here for quick reference.

Tables 6.1 and 6.2 show the official SENSEVAL scores for both k-NN and SVM based implementations for the best feature based clustering system (chosen using the performance on development data set). For easy comparison, the tables also provide the baseline results, and the performance of the system which used WORDNET lexicographer files as sense classes (cf. tables 4.4 and 4.5). In both classifier settings, the end systems that use feature based clusters can outperform the systems that use WORDNET lexicographer files.

The following sections will cover only the k-NN classifier results, which is our primary interest and yields better performance.

6.7.2 Reduction in Sense Loss

One trivial advantage that was mentioned about a finer set of sense classes is that they provide a better resolution in terms of fine grained senses. The concept of sense loss was formulated in section 2.5; this is the proportion of labeled sense instances in a corpus that lose their original sense due the fact that a given class can retain only one fine

grained sense in the backward assignment from class to sense. Figure 1.2 showed the proportion of senses lost for polysemous senses in SEMCOR, and figure 1.3 showed the loss of accuracy in actual SENSEVAL data sets due to sense loss. Although we argued that this loss is affordable even at a coarse level of lexicographer files, it is possible to reduce this loss by using a finer set of classes. On the other hand, the smaller the individual classes, the smaller the amount of generalization, and consequentially, smaller the amount of available training data per class. This is an undesirable effect. So it is useful to analyze how the size of classes affect the reduction in sense loss. We will analyze this for SEMCOR corpus and SENSEVAL English all-words task training data.[1]

Figures 6.3 and 6.4 show the sense loss in SEMCOR corpus as well as SENSEVAL English all-words task data for different numbers of clusters, both for purely feature based (section 6.3) and WORDNET hierarchy segmented (section 6.4) clusters.

It is intuitive to assume that as the number of clusters increase, the sense loss decreases. This can be expected as a large number of classes implies smaller size per class, hence reduced probability of more than one sense of the same word falling into the same class.

The starting point of the number of classes, both for nouns and verbs, is the original number of respective lexicographer files. In the case of WORDNET hierarchy segments, a sharp drop is visible when we change the number of classes from this original number to the next step. This shows that the lexicographer files tend to group several senses of the same words into the same class, and even a small flexibility can break this kind of clumping, when there are enough labeled instances per sense to figure out the differences between senses. Only senses with significant proportional frequencies are the ones that can give reliable clues for efficient clustering; still, it is beneficial to focus on at least these senses, as this kind of frequent senses are the ones that will contribute to a large sense loss (in the place of secondary sense of a class).

After this initial loss in WORDNET hierarchy constrained clustering, the drop in sense loss is somewhat linear with the number of classes, which is something that can

[1] Although SENSEVAL lexical sample tasks provide larger sets of training data, they do not provide a reliable estimation because the number of words involved are limited, and even within that sample, there is no guarantee that the distribution of senses reflect the distribution in real-world text.

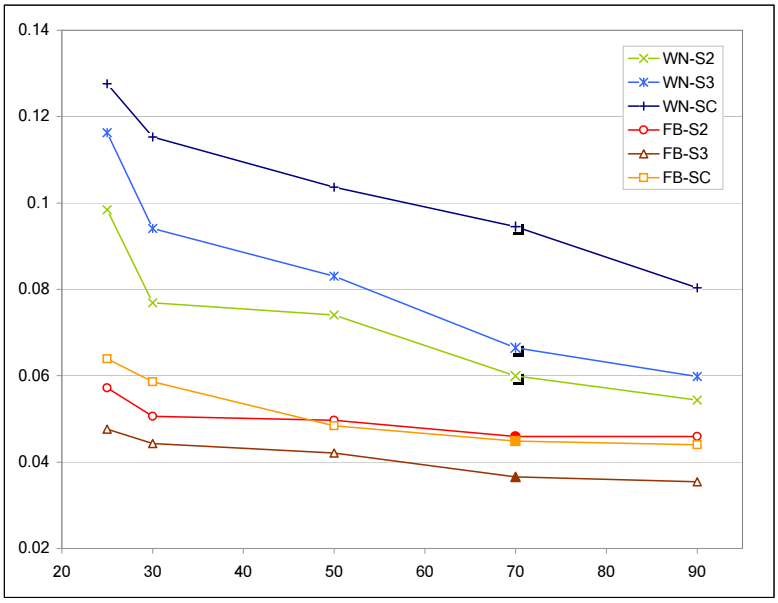

Figure 6.3: Sense loss with varying cluster sizes for nouns. Proportional sense loss against the number of clusters. WN: WORDNET tree segments, FB: feature-based clusters. S2 and S3 are SENSEVAL-2 and 3 English all-words task test data respectively. SC is SEMCOR labeled data The optimal setting for clusters (determined by classifier performance on development data) is highlighted.

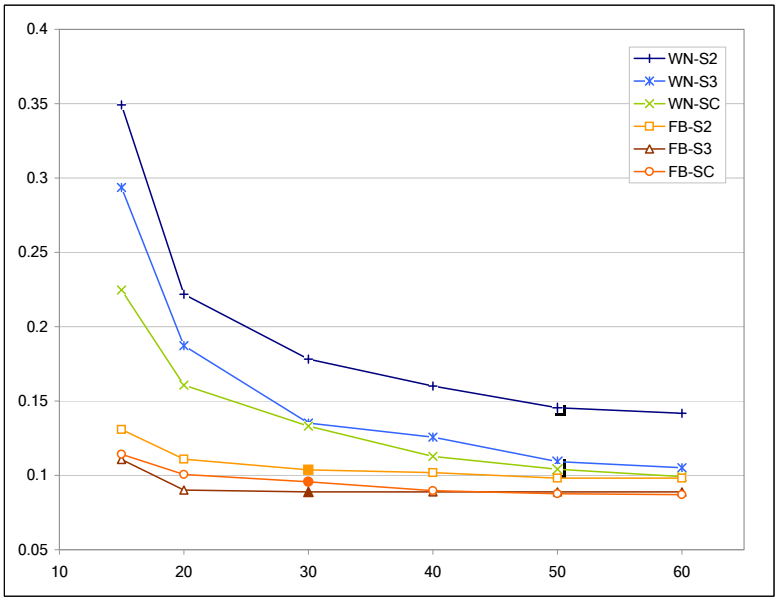

Figure 6.4: Sense loss with varying cluster sizes for verbs. Details as per figure 6.3; However, the improvements over smaller cluster sizes in feature based classes is much clear.

be intuitively expected. However, the behavior of loss in feature based clusters is much different: even at small numbers of clusters, they have very small sense loss. This is not something that could be easily anticipated, as the initial design did not specifically focus on separating the senses of the same words into different clusters. What was expected was that the larger number of classes will naturally result in a reduction in sense loss, *gradually*, as the number of classes increase. However, this behavior, though unexpected, is advantageous for the reasons mentioned above: Larger classes mean more training examples per class, hence are desirable.

Though unanticipated, it is not hard to explain the reason for this improvement. The WORDNET hierarchy based clustering scheme relies on tree segmenting in order to reduce the total variance, and it does not allow interchange of senses among classes. Each split increases the number of classes, while reducing the total variance by some extent. The degree of freedom is limited to determining which node to split at, in each round. On the other hand, feature based clustering scheme can choose to simply rearrange the senses among classes in order to reduce the variance, and it selects splitting as only a way of handling extreme outliers. This provides a more efficient way of handling differences within clusters, as a very large number of rearrangements is possible without increasing the number of clusters, providing much greater degrees of freedom.

Another reason to support this explanation is the apparent flat nature of the curves for feature based clusters. It is evident that the increase in the number of classes does not decrease the sense loss much, because the gains from rearrangements at initial steps is substantially larger compared to gains over splitting.

Which senses fall into which clusters depend purely on the usage and the discretion of lexicographers, who determine that a particular usage merits an individual sense assignment. This latter factor has a strong implication on the sense loss, as it is theoretically possible that the variations of 'sub' senses that lexicographers deem fit for separation are not identifiable in the corpus, in terms of textual features we use (or in worse case, the different sub senses have *similar* features in the way we formulated our features). If this had been the case in reality, the automatic clusters would have ended up even increasing the sense loss rather than decreasing it. But our observations above

		SENSEVAL-2	SENSEVAL-3
noun	WORDNET LFs	90.6%	89.5%
	Feature based classes	**96.6%**	**96.8%**
verb	WORDNET LFs	63.1%	69.5%
	Feature based classes	**89.7%**	**91.0%**

Table 6.3: Reduction in sense loss on correctly classified coarse-grained instances, in actual SENSEVAL answers. Listed values are the proportions of the answers that were mapped into correct fine-grained sense, out of the instances that were labeled with the correct class. Results are listed for WORDNET lexicographer files (cf. tables 4.16 and 4.17) and feature based sense classes.

show that this is not the case fortunately; significant secondary senses that warrant separation (as decided by the lexicographer) do seem to have statistical support even in a corpus like SEMCOR , which is hopelessly tiny compared to those employed by real-world lexicographers. This is even more interesting because WORDNET senses are not originally derived from corpus statistics.

Table 6.3 shows the new reduced sense loss values, along with the previous values when WORDNET lexicographer files are used (cf. tables 4.16 and 4.17), for correctly classified instances at the class-level. Since the classes based on feature-based clusters are large in number, and since there are no meaningful 'labels' attached to them, as was the case of WORDNET lexicographer files, it is not useful to list per-class sense loss. The method used for calculating the loss value is the same as in previous case; The values given are the sense 'coverage' values; to recall from section 4.8.1,

$$\text{coverage} = \frac{\text{number of instances where predicted fine grain sense was correct}}{\text{number of instances where predicted class was correct}}$$

and shows how much the given class map could have contributed to the incorrect answers by inherent lack of some senses that are found in the answer keys. Comparison with loss in WORDNET lexicographer files with that of the sense map based on feature based clusters show that the sense loss is considerably reduced in the actual answer keys when feature based sense map is used.

		SENSEVAL-2		SENSEVAL-3	
		fine	coarse	fine	coarse
noun	WORDNET LFs	0.740	0.817	0.728	0.813
	Feature based classes	0.755	0.782	0.741	0.765
verb	WORDNET LFs	0.453	0.718	0.552	0.794
	Feature based classes	0.473	0.522	0.567	0.623

Table 6.4: Fine and coarse grained performance compared. As the number of classes grow, fine and coarse grain performance converge to some extent.

6.7.3 Coarse Grained and Fine Grained Results

Previous section (6.7.2) analyzed the effect of sense loss from the perspective of *correctly labeled* instances. This is the actual figure that matters in the class to fine-grained sense mapping; however, a comparison of the classifier performance at fine-grained and coarse-grained levels is more convenient in analyzing the big picture. This is given in table 6.4.

The coarse grained performance of the system that uses WORDNET lexicographer files is much better compared to the coarse grained performance of the system that uses the feature based classes. This is due to the fact that WORDNET lexicographer files are quite large. The finer the classes, the closer the coarse-grained result to fine-grained results. If 'coarse grain' classes were made to be the finest possible case (where each class has only one sense), the two performance figures will obviously converge. This does not mean that the performance figure at this particular point is necessarily the best fine grained performance; larger size of the clusters, in general, increase both classifier performance and sense loss. Increased coarse grained performance figure at smaller numbers of classes is partially due to the fact that classes are large, rather than they are easier to learn. Large classes also result in poor sense loss figures, and thus poor fine-grained sense performance. The setting of clusters that perform best on the fine-grained senses lie somewhere midway, and there is no straightforward relationship, based on theoretical factors, which we can use to find this number. This justifies the empirical approach of selecting clusters on their fine-grained performance on development set.

6.7.4 Improvement in Feature Information Gain

Another important feature in terms of classifier accuracy is how well the classes can be separated by the set of features that is being used. For this, we will use an objective measure: feature information gain.

Information gain measures how well a given set of classes can be separated by a given feature. It is an information theoretic measure, defined in the following way.

Assume that a set S of instances have a set of classes $C = c_1, c_2, c_3, \ldots, c_n$, and each class has a distributional probability $p(c_i) = N(c_i) / \sum_{j=1}^{n} N(c_j)$. Then, the system entropy $H(C)$ is defined as

$$H(S) = \sum_{i=1}^{n} -p(c_i) \cdot \log p(c_i)$$

Suppose a feature with values V is used to divide the set of classes into different subsets; if the feature was good enough, when it separates the instances into different groups depending on different feature values, each group thus separated will have zero entropies with respect to the set of classes. Formally, we define the information gain of the feature as the difference between the original system entropy, and the weighted (according to the size of each group) sum of the entropies of the resulting systems after the separation. In other words, the information gain of a feature f_k with a set of values V_k is

$$G(f_k) = H(S) - \sum_{v \in V_k} \frac{|S_v|}{|S|} \cdot H(S_v)$$

where S_v is the subset of instances for which f_k has the value v. With an optimal separation, the latter sum will be zero, because a given feature value will belong to only one class of instances. If the feature has no information whatsoever, the entropies of the subsets will be similar to that of the complete set. In between these two ends, the measure gives an idea how 'clean' the separation is.

TiMBL can report the information gain for each feature used in the training file. This option was used to calculate the information gain for the $[-3, +3]$ feature win-

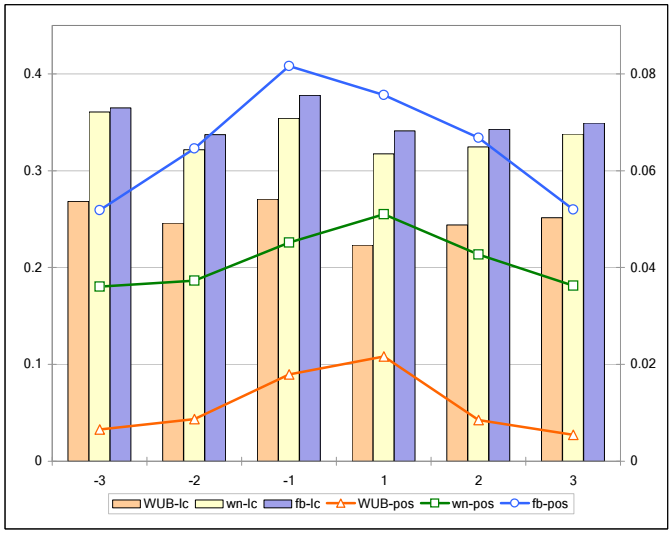

Figure 6.5: Improvement on Information Gain for Different Clusterings for nouns in SEMCOR labeled data. WUB - WORDNET lexicographer files, wn- and fb- are WORDNET tree split and feature based clusterings. lc and pos are for local context and parts of speech based clusters.

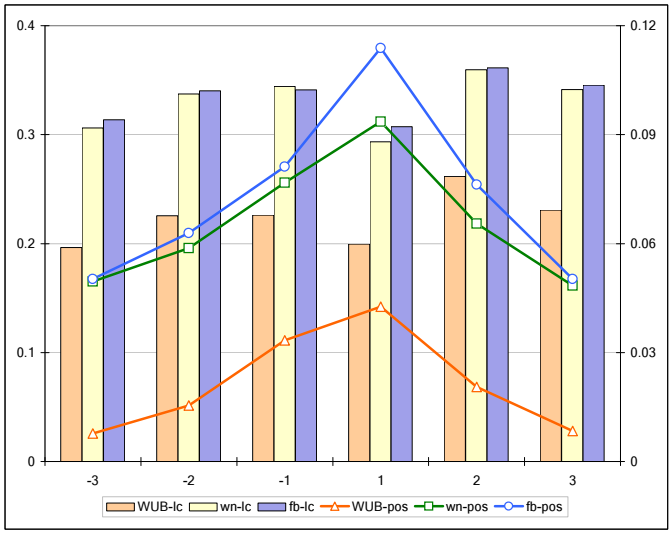

Figure 6.6: Improvement on Information Gain for Different Clusterings for verbs in SEMCOR labeled data. WUB - WORDNET lexicographer files, wn- and fb- are WORDNET tree split and feature based clusterings. lc and pos are for local context and parts of speech based clusters.

	noun	verb	adj.	total
baseline	0.711	0.439	0.639	0.623
WORDNET lex files	0.724	0.455	0.639	0.634
WORDNET partition, POS	0.724	0.453	0.639	0.633
WORDNET partition, local context	0.723	0.457	0.639	0.633
feature based, POS	0.725	**0.480**	0.643	0.642
feature based, local context	**0.747**	0.458	**0.654**	**0.650**

Table 6.5: Results for different clustering schemes in SENSEVAL-2 English all-words task data, with simple majority voting.

dow for both parts of speech and local context features, for different sets of classes: original WORDNET lexicographer files, feature based classes, and WORDNET hierarchy segments. Note that the features of the instances do not change over different class schemes; only the possible set of classes each instance is assigned gets changed. The better gain in information hints that a given set of classes is better separated by the given set of features.

6.8 Results in SENSEVAL Tasks: Analysis

Earlier in section 6.7.1 (tables 6.1 and 6.2) the results were reported for feature based clustering on SENSEVAL tasks. This was for final weighted-voting classifier, with the best options and voting weights found using the development data set. Here, the results for each individual scheme are provided. Recall that there are two clustering systems: strictly feature-based system which was our proposal, and WORDNET tree segmenting scheme which was the control experiment. Both systems were clustered using both local context and POS features, resulting in four systems, which are compared here with the original WORDNET lexicographer file scheme, which was described in section 4.7.

Tables 6.5 and 6.6 show the results with simple majority voting for SENSEVAL tasks.

6.8.1 Effect of Different Class Sizes

Tables 6.7 and 6.8 respectively show the performance levels of nouns and verbs at different numbers of classes in actual SENSEVAL tasks, using simple majority voting. The

	noun	verb	adj.	total
baseline	0.700	0.534	0.669	0.634
WORDNET lex files	0.719	0.548	0.669	0.647
WORDNET partition, POS	0.717	0.559	0.669	0.650
WORDNET partition, local context	0.719	0.557	0.669	0.651
feature based, POS	0.710	**0.568**	0.694	0.655
feature based, local context	**0.736**	0.541	**0.708**	**0.659**

Table 6.6: Results for different clustering schemes in SENSEVAL-3 English all-words task data, with simple majority voting.

Num. of classes	SENSEVAL-2	SENSEVAL-3
30	0.732	0.728
50	0.727	0.716
70	0.747	0.736
90	0.752	0.711

Table 6.7: Performance at different numbers of classes: nouns.

Num. of classes	SENSEVAL-2	SENSEVAL-3
20	0.455	0.567
30	0.480	0.568
40	0.480	0.572
50	0.451	0.561
60	0.453	0.560

Table 6.8: Performance at different numbers of classes: verbs.

	noun	verb	adj.	total
baseline	0.711	0.439	0.639	0.623
WORDNET lex files	0.740	0.453	0.639	0.641
WORDNET partition	0.713	0.458	0.639	0.629
feature based	0.755	0.473	0.649	0.657

Table 6.9: Results with weighted voting for different clustering schemes in SENSEVAL-2 English all-words task data

	noun	verb	adj.	total
baseline	0.700	0.534	0.669	0.634
WORDNET lex files	0.728	0.552	0.669	0.653
WORDNET partition	0.720	0.555	0.669	0.650
feature based	0.741	0.567	0.691	0.668

Table 6.10: Results with weighted voting for different clustering schemes in SENSEVAL-3 English all-words task data

numbers selected from the results of the development data set (70 for nouns and 30 for verbs) do fairly well for both parts of speech, although it can be seen that there are some clusterings that perform slightly better than this on SENSEVAL data, for instance 40 classes in verbs. With a proper validation set issues like this could have been minimized, but it remains a question whether such a validation set can always be found.

6.8.2 Weighted Voting

The weighted majority algorithm, which significantly increased the performance in using WORDNET lexicographer files, could yield similar improvements of results in this systems as well. In particular, it improved the results of the feature based cluster system significantly.

Recall that for each cluster system (WORDNET tree segmenting and purely feature based clusters), there were two clustering schemes for each part of speech, one based on parts of speech features and one on local context features. In order to simplify the matters, the scheme that yielded the best performance in development data set for each part of speech (POS-based clusters for verbs, local context based clusters for nouns and adjectives) was used in weighted voting.

The results after voting are shown in tables 6.9 and 6.10. The entries 'WORDNET lex files' show the previous result of weighted voting on the classifier using WORD-

NET lexicographer files as generic classes; 'WORDNET partition' shows the results for WORDNET hierarchy partitioning based sense classes, and 'feature based' is the scheme where clusters based on contextual features are used. This scheme yields the best performance.

6.8.3 Statistical Significance

The tests showed that the significance levels of the results are somewhat limited. Not many previous work had reported detailed statistical significance figures on the data sets we used in these experiments, hence comparing with previous results is not convenient.

McNemar's test was used for calculating the statistical significance of the performance differences. SENSEVAL official scorer was used in verbose mode so that it reports the results for each instance individually, and the accuracies were compared pair-wise.

Tables 6.11 through 6.16 show the significance tables for simple majority voting results, for nouns, verbs and total in SENSEVAL-2 and 3 data.

Significance patterns for the results with weighted voting are shown in tables 6.17 and 6.18.

	TP-P	TP-LC	WLF	FB-P	FB-LC
T-PP	*	$\prec$	$\prec$	$\prec$	$\ll$
TP-LC		*	$\prec$	$\prec$	$\ll$
WLF			*	$\prec$	$\ll$
FB-P				*	$\prec$
FB-LC					*

Table 6.11: Significance figures: SENSEVAL-2 complete results - simple voting. $\ll$ and $\gg$: $p \leq 0.01$; $<$ and $>$: $0.01 < p \leq 0.05$; $\prec$ and $\succ$: $p > 0.05$. Entries starting with TP: WORDNET hierarchy tree partitioning, entries starting from FB: feature-based sense clustering, WLF: WORDNET lexicographer file based system. Suffixes -P and -LC denote POS and local context based schema respectively.

	TP-P	TP-LC	WLF	FB-P	FB-LC
TP-P	*	$\succ$	$\prec$	$\prec$	$\ll$
TP-LC		*	$\prec$	$\prec$	$\ll$
WLF			*	$\prec$	$\ll$
FB-P				*	$\ll$
FB-LC					*

Table 6.12: Significance figures: SENSEVAL-2 nouns - simple voting

	TP-P	TP-LC	WLF	FB-LC	FB-P
TP-P	*	$\prec$	$\prec$	$\prec$	$\ll$
TP-LC		*	$\succ$	$\prec$	$\ll$
WLF			*	$\prec$	$\ll$
FB-LC				*	$<$
FB-P					*

Table 6.13: Significance figures: SENSEVAL-2 verbs - simple voting

	TP-P	TP-LC	WLF	FB-P	FB-LC
TP-P	*	$\prec$	$\prec$	$\prec$	$\prec$
TP-LC		*	$\prec$	$\prec$	$\prec$
WLF			*	$<$	$\ll$
FB-P				*	$\prec$
FB-LC					*

Table 6.14: Significance figures: SENSEVAL-3 complete results - simple voting

	TP-P	TP-LC	WLF	FB-P	FB-LC
TP-P	*	$\succ$	$\prec$	$\prec$	$\ll$
TP-LC		*	$\prec$	$\prec$	$\ll$
WLF			*	$\prec$	$\ll$
FB-P				*	$\ll$
FB-LC					*

Table 6.15: Significance figures: SENSEVAL-3 nouns - simple voting

	TP-P	TP-LC	WLF	FB-LC	FB-P
TP-P	*	$\succ$	$\succ$	$\succ$	$<$
TP-LC		*	$\succ$	$\succ$	$<$
WLF			*	$>$	$\prec$
FB-LC				*	$\ll$
FB-P					*

Table 6.16: Significance figures: SENSEVAL-3 verbs - simple voting

	all				**nouns**				**verbs**		
	WTP	WLF	FB		WTP	WLF	FB		WLF	WTP	FB
WTP	*	$<$	$\ll$	WTP	*	$\ll$	$\ll$	WLF	*	$\prec$	$<$
WLF		*	$<$	WLF		*	$\ll$	WTP		*	$<$
FB			*	FB			*	FB			*

Table 6.17: Significance patterns for weighted voting schemes: SENSEVAL-2 data. $\ll$ and $\gg$: $p \leq 0.01$; $<$ and $>$: $0.01 < p \leq 0.05$; $\prec$ and $\succ$: $p > 0.05$. WTP: WORDNET hierarchy tree partitions, WLF: WORDNET lexicographer files, FB: feature-based sense clustering.

	all				**nouns**				**verbs**		
	WTP	WLF	FB		WTP	WLF	FB		WLF	WTP	FB
WTP	*	$\prec$	$\ll$	WTP	*	$\prec$	$<$	WLF	*	$\prec$	$\ll$
WLF		*	$\ll$	WLF		*	$\ll$	WTP		*	$\prec$
FB			*	FB			*	FB			*

Table 6.18: Significance patterns for weighted voting schemes: SENSEVAL-3 data.

	noun	verb	adj	total
SENSEVAL-2	0.749	0.471	0.639	**0.651**
SENSEVAL-3	0.730	0.564	0.669	**0.658**

Table 6.19: SVM-based system results

6.8.4 Support Vector Machine Implementation Results

As was the case with support vector machine implementation for WORDNET lexicographer files (see section 4.9), results of SVM-based implementation for feature based classes do not differ much from the results for k-NN classifier based implementation. The result figures are reported in table 6.19 for all parts of speech and in total. The final results, where adverbs and multiple word phrases were added to this, was given in section 6.7.1 (tables 6.1 and 6.2) for easy comparison with the same results of other systems.

Comparing with the result for WORDNET lexicographer files (cf. table 4.25) It is evident that the feature based clustering algorithm performs better than original WORDNET lexicographer files with SVM classifier as well.

6.9 Syntactic Features and Taxonomical Proximity

One reason that the automatically generated classes enhance the performance of the classifiers is that the feature based classes provide a grouping of senses independent of the taxonomical hierarchy. This enables better utilization of the information contained in Jiang and Conrath similarity measure, because the new classes provides information *complementary* to what the hierarchy implicitly provides to the classification algorithm via Jiang and Conrath measure. This was not the case with WORDNET LFs.

For instance, one small cluster in local-context-based clustering had the concepts *kneecap/1*, *forearm/1*, *palm/1*, *coattail/1*, *overcoat/1*, *shirtsleeve/1*, *homeland/1*, and *motherland/1*. Clearly, the concepts fall into three different groups, i.e. body parts, clothing parts and places.

The context window feature, as it is represented in the text form, does not directly correspond to the values of the coordinate vector resulting from SVD. Still, it is possi-

kneecap/1	chin on **his** stretching **his** neck
forearm/1	bowed shoulders **his** across his knees
palm/1	NULL place flat on either side
palm/1	on **his** lap up stiffly motionless
palm/1	slick between **his** and the stick
coattail/1	tucked under **his** and staring into
overcoat/1	a new spring and a taffy
overcoat/1	**his** gray tweed and **his** city
overcoat/1	hatless in an of rough blue
shirtsleeve/1	second twitched **his** and he felt
homeland/1	longing for **his** and its boyhood
homeland/1	to the wintry of **his** fathers
motherland/1	NULL in **his** in the spacious

Table 6.20: Different conceptual groups in a single contextual cluster. The six word context window surrounding the instances in the labeled data set. At least one common feature (his) can be manually observed.

ble to identify one single feature, word *his* in context, albeit at different positions. This shows that the contexts of all cases have at least some visible similarity, and it is possible that SVD picked up some latent relationships among the contexts. The smoothing technique of averaging the instances of all senses would also have helped for this, by compensating the different positions of placement of clue words.

This manner of clumping senses together may seem like undesirable, as these commonalities do not seem to create a meaningful class of senses. However at the classifier level, this matter is more or less taken care of by the similarity weighting of examples. The Jiang and Conrath measure (section 2.3.1) is strongly based on the semantic hierarchy of WORDNET. This imposes a soft constraint on the classifier, making sure that training instances with senses that fall within the close taxonimical proximity have a better influence on test instances. For instance, *kneecap/1* has similarities about 0.06 to *forearm/1* and *palm/1*, but only about 0.04 similarity values to all other senses. Similarity between *shirtsleeve/1* and *overcoat/1–coattail/1* is about 0.8, while other unrelated senses have only around 0.4 similarity values to *shirtsleeve/1*. Final outcome of this is that the feature based classes achieve both syntactic and semantic coherence, and avoids both semantically meaningless 'examples' as well as Wierzbicka-style (Wierzbicka, 1996) discrepancies on the part of purely taxonomical nature of the WORDNET hierarchy.

Opposition to this observation can come from the issue that the previous class system *does* seem to have the same trait in a more direct manner, as the classifier calculates nearest neighbors in terms of features, while the similarity weighting restricts the influence on the test instance by individual training instances. It can be argued that differences in contextual features will automatically make distant instances when Wierzbicka-style differences are found within text, even for taxonomically close examples. If this happens, a class system based on the same features could have been thought of as redundant. However, it should be noted that the classifier is only concerned about individual instances, while the clustering system works on averaged and filtered (through SVD) set of features. Because of this, the influence of classes, although not as direct as the inter-instance distance, provides a more smoothed out and reliable version of information. Also, the inter-instance distance information is still available to the classifier, and can be used the same way as in the lexicographer file based system.

6.10 Summary

The ultimate set of sense classes would facilitate optimal feature-based learning, by making sure that the set of features used in learning provides consistent clues for classes. Sense classes formed this way will be cohesive in terms of features, with maximum similarity of features within classes, and minimal similarity between classes. This chapter presented a system that was designed to achieve this goal of contextual coherence through clustering techniques. In addition to this, a control experiment was conducted, where the formation of classes was constrained to the WORDNET hierarchy in terms of sense placement, but classes had sense groups smaller than WORDNET LFs. This was done in order to verify whether the class size is the only factor that affects cohesion.

Results of both systems show that classes based on contextual features alone yield better performance with the help of semantic similarity measures at classifier phase. This confirms our idea that the ideal generic class system for automatic WSD should take into account the contextual features as well as taxonomical semantics.

Chapter 7

Sense Partitioning: An Alternative to Clustering

The approach this thesis suggests as a solution to data sparsity in WSD is to learn generic classes of word senses, which helps increase the amount of training instances available for each word. Previous chapters discussed one technique we suggested for learning generic classes, and discussed how to generate a set of classes that are end-task oriented – a kind of classes that helps increase the final classifier accuracy in generic sense class learning. These classes were generic among word senses, that is, the same set of classes were shared between a group of different words. In this representation, if sense $s_{i,x}$ of word x is in the same class with sense $s_{j,y}$ of word y, then $s_{i,x}$ can be a substitute example for $s_{j,y}$, and sense $s_{j,y}$ for sense $s_{i,x}$. In other words, the 'same class' relationship is *reflexive*.

This assumption is essentially a constraint on the relations among senses, as it can force the senses to be in the same class even when they are not necessarily close, as long as putting them together increases the overall class quality: The fact that sense $s_{j,y}$ is among the nearest n senses for sense $s_{i,x}$ does not necessarily mean that sense $s_{i,x}$ is also among the nearest n neighbors of sense $s_{j,y}$. In other words, while $s_{j,y}$ may be a suitable substitute example for $s_{i,x}$, sense $s_{i,x}$ might not be among the best substitutes for $s_{j,y}$. One way to relax this criterion to make the 'substitute' relationship *non-reflexive* – to define the word senses that are closer to a given word sense as belonging to the same class as the latter sense, without constraining ourselves to assume that the converse

also holds. This relaxation generates a per-word clustering without trying to maximize *overall* sense cluster quality of a given lexicon.

In this chapter, a method will be presented that tackles this problem: the basic arrangement used here is similar to the previous experiments, as this method also works on the coordinates made of principal components of contextual features of word senses. However instead of clustering all instances at the same time, we use a method of partitioning senses *per word*. Since the partitions are different for different words, separation of senses per individual word does not constrain the algorithm too much for a 'global' optimum.

In effect, this method has all the desirable features of a previous classifier framework; it provides a way to find substitute training examples for senses according to its usage. The same framework of that was used in the previous experiments can be used for training and classifying with minor modifications. However, this method eliminates two undesirable features of the previous system. First it reduces the possibility of having low-quality relationships among individual senses, which can arise from the fact that generic classes do not optimize instances for individual words, but the overall class quality. Second, it brings the sense loss to a theoretical zero, as we will discuss in the next section.

7.1 Partitioning Senses Per Word

The idea behind partitioning is intuitive, and is best explained by Voronoi Diagrams (Aurenhammer, 1991), a well-known data structure in computational geometry.

Let us assume, following the familiar notion (from section 6.2), that the principal components of the feature vector for the senses form the set S of points in $\mathbb{R}^d$. Each word sense is a point in this space. For a given word w_i with senses $S_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,n}\}$, we define the *dominance* of sense $s_{i,j}$ over sense $s_{i,k}$ as

$$dominance(s_{i,j}, s_{i,k}) = \{x \in S \mid \delta(x, s_{i,j}) < \delta(x, s_{i,k})\}.$$

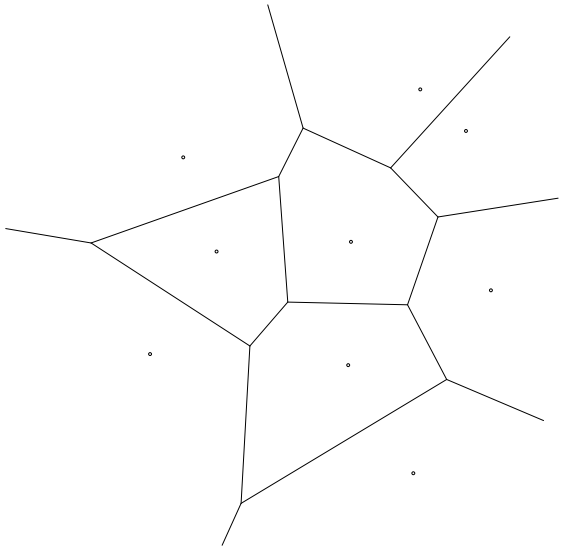

Figure 7.1: A sense partitioning. Dots denote 'centers' of partitions, i.e. the senses of the word considered.

where $\delta(x, y)$ is the Euclidian distance between cordinates corresponding to the senses x and y. The *region* occupied by a given sense $s_{i,j}$ of w_i is the set intersection of all dominance sets of $s_{i,j}$.

$$region(s_{i,j}) = \bigcap_{s \in S_i - si,j} dominance(s_{i,j}, s)$$

Simply put, region of a sense $s_{i,j}$ is the set of all senses (essentially, of other words than w_i) that lie closer to $s_{i,j}$ than to any other sense of w_i. After forming the partition we can select the substitutes that are near to the 'center' senses. For each word, we repeat this process.

In a way, this is essentially similar to a single-step in k-means algorithm we used earlier. However in this approach, the centers are exact senses of the word considered; a partition, unlike a cluster in previous experiments, does not include more than one sense of the word for which the partitioning was done. Because of this, the separation of senses, for each word, is at an *a priori* optimum, as the sense loss in the system is reduced to a theoretical zero. Creating partitions in this way can be thought of as just an intuitive next step of the same calculation steps done in the previous experiments. Of course, if there were no instances in SEMCOR for a particular sense, it will not be

used in partitioning; however, this does not introduce any new problem, as the same condition means there are no training instances for that particular sense even in a standard classification task; no target instance will be classified into a sense for which there were no training instances.

After distributing all senses among partitions, two methods can be used to pick substitute senses from the partitions. First is to select all senses that fall within a certain radius from the center senses. Second is to pick a fixed number, or a fixed proportion, of senses that are nearest to the center. However, the coordinate distribution of the senses is such that different partitions have different neighbor densities. This means that the number of senses that fall within a given radius can considerably vary from partition to partition. Still, this approach of including all senses within a given proximity has one advantage: In a case where a neighborhood of a center is rather dense, picking the nearest n number of senses can favor totally unrelated senses that happens to lie (due to some spurious feature) slightly nearer to the center, rather than a sense which has a considerable semantic relatedness. Picking a fixed portion (instead of a fixed number) of senses that are nearest to the center also suffers from the same issue. On the other hand, the approach of picking all instances within a fixed radius to the center is not practical either. Problem with this approach is that it has too many unknowns to work with; for instance, the variance of partitions differ dramatically, and a fixed radius from center that includes all instances in one partition could include no instances from another partition, if none of its senses fall within the said radius to its center. For these reasons it is more prudent to employ an approach of selecting all senses within a radius that varies depending on the partition variance. This addresses both problems mentioned above: Because all instances within a given radius is selected, it is less likely to discard relevant but slightly distant instances, given the radius of selection is large enough. Since the radius is dynamically adjusted, it ensures that some neighbor instances are selected for every sense, but not too many instances are selected either.

7.1.1 Classifier System

Once the sense partitions are created, they can be used in the same system as which was used in the previous experiments, with only small structural modifications. The most important change is that this system does not have a class map, but works directly on fine-grained senses; this avoids the problem of sense loss altogether. Instead of gathering substitute examples from members of the same sense class, as was done in previous experiments, one can use 'neighbor' senses for each word sense to generate substitute examples. After this, classification can directly proceed at fine grained sense level.

Feature vectors, similarity weighting, and the classifier can be the same as the previous experiments, and the system can use the weighted majority algorithm for final classifier voting.

7.2 Neighbor Senses

Many unrelated senses which have similar syntactic patterns can be selected as candidates for substitute senses in a partition. This is not different from the observation discussed in section 6.9. As was in the previous experiments, it is still possible to use a measure of semantic similarity, and to determine which examples are actually 'authoritative' and which are not. In practical implementation, the same similarity measure (Jiang and Conrath: see section 2.3.1) was used as in the previous cases.

For instance, consider the following instances that were selected as nearest neighbors for noun sense *study/1* - 'A detailed critical inspection':

> *analysis/1 concentration/5 concern/1 design/2 detail/1 *director/1 discussion/1 *editor/1 evidence/2 examination/1 investigation/1 knowledge/1 *manager/1 measurement/1*[1], **member/1 objective/1 observation/1 question/2 *teacher/1 understanding/1*.

[1] WORDNET 1.7.1 lists 'the act or process of measuring' as the only sense of *measurement*, which compares closely with *study/1*, 'A detailed critical inspection'. Also included is the sense *observation/1* 'the act of making and recording a measurement', not the actual observation made. All of these senses belong to the same lexicographer file ACT as well.

Marked with '*' are five words that do not have any perceivable relationship with sense *study/1*. However, the conceptual similarity they have among themselves show that the inclusion of this kind of 'spurious' senses must be because of some unrelated contextual reason, rather than purely random: sense instances in this particular example, as it happens, share similar collocation patterns in the SEMCOR corpus, for instance the local context vector `<NULL NULL the * of>` where `*` denotes the candidate word. As singular value decomposition step does not take semantic similarity into account, it is natural that the four-token frequent pattern is regarded as a very significant one. As mentioned earlier, similarity weighting helps solve this problem, by complementing the contextual feature based information with taxonomical constraints.

One other interesting, but not much surprising, fact is that the antonyms of words are selected as the best substitute senses, as well as synonyms. This is not a fault of partitioning algorithm, as antonyms of a sense typically behave the same way the original sense does. For instance, the nearest sense set for verb sense *abandon/1* is

> *accept/2 adopt/1 alter/1 attend/1 change/2 consider/2 create/1 destroy/2 develop/1 eliminate/1 encourage/1 end/2 enter/3 establish/1 face/1 face/2 finance/1 furnish/1 get/1 handle/1 have/6 hold/2 improve/1 increase/1 increase/2 join/1 justify/1 keep/1 locate/1 maintain/1 make/1 match/1 meet/5 obtain/1 operate/1 perform/1 permit/2 preserve/1 prevent/1 promote/1 protect/1 provide/1 raise/1 reduce/1 remove/1 secure/1 sell/1 sing/1 strengthen/1 supply/1 undertake/1 visit/1 win/1*.

Half of this list is made of verbs that have just the opposite semantics of *abandon/1*. It can be seen (assuming antonyms share similar contextual features) that the differences within senses of the same are somewhat vague at contextual level as well. For instance, the second sense of *abandon* is 'stop maintaining' and 'maintain' is an entry for *abandon/1*, showing the contextual similarity. Another similar entry is *keep/1* which is the opposite sense of *abandon/3* (give with the intent of never claiming again).

System	Recall
Baseline (WORDNET first sense)	0.658
WORDNET LFs: k-NN	0.674
Feature based class map: k-NN	0.687
Partitioning based samples	**0.689**

Table 7.1: Final results of partitioning based sampling on SENSEVAL-2 data. For comparison, baseline and previous best results using feature based clusters are also given.

System	Recall
Baseline (WORDNET first sense)	0.643
WORDNET LFs: k-NN	0.661
Feature based class map: k-NN	0.677
Partitioning based samples	**0.683**

Table 7.2: Final results of partitioning based sampling on SENSEVAL-3 data.

7.3 WSD Results

As the classifier methodology is not different from the previous experiments except for the differences discussed above, this section will directly proceed into presenting the results of the system in SENSEVAL-2 and SENSEVAL-3 evaluation data.

As earlier, we will first present the results for SENSEVAL tasks, including all parts of speech and multiple-word expressions. These are shown in tables 7.1 and 7.2. There is a small improvement over the sense classes based on features, which was discussed in section 6. These results are the best presented in this work, although the performance improvement over the previous best system (usage based sense clusters) is not statistically significant.

Tables 7.3 and 7.4 show the result for both SENSEVAL tasks for different parts of speech, for both simple majority and weighted majority voting. The patterns of the weighted voting scheme is similar to those of previous experiments, nouns yielding the best performance improvement.

	noun	verb	adj.	total
baseline (sense 1)	0.711	0.439	0.639	**0.623**
simple majority voting	0.739	0.469	0.662	**0.650**
weighted majority voting	0.754	0.471	0.667	**0.659**

Table 7.3: Results of sense partitioning system in SENSEVAL-2 test data. Results are provided for three parts of speech, and both voting schemes.

	noun	**verb**	**adj.**	**total**
baseline	0.700	0.534	0.669	**0.634**
simple majority voting	0.730	0.571	0.696	**0.666**
weighted majority voting	0.742	0.577	0.702	**0.674**

Table 7.4: Results of sense partitioning system in SENSEVAL-3 test data. Results are provided for three parts of speech, and both voting schemes.

7.4 Summary

Sense loss was shown to be a significant source of errors in the generic sense class based WSD systems. The ultimate reduction in sense loss requires that each class contains at most one sense from each word; constraining the sense clustering algorithm to achieve this end result is not practical. The alternative presented here uses a partitioning technique, which held fine grained senses of a given word as centers, while allowing the senses of other words to be partitioned around them. This reduces the sense loss to zero, while providing more cleaner clusters, and better performance.

Chapter 8
Conclusion

In this thesis, an alternative approach to the problem of word sense disambiguation was investigated .

The classic Word Sense Disambiguation agenda had traditionally been based on two major groups of techniques: supervised and unsupervised. Supervised learning generally yielded considerably good results; but they assume, almost always, a classic model of supervised machine learning, where the availability of adequate amounts of labeled training data is taken for granted. This is not always the case for WSD, as manually labeling data with word sense tags per each word is prohibitively expensive. Many would agree that the most serious problem faced by contemporary word sense disambiguation is that of the knowledge accusation bottleneck, or obtaining enough training data for supervised learning. There have been various approaches that tried to solve this problem through unsupervised learning: unsupervised techniques are particularly appealing in the contemporary setting, where very large amounts of text are available in machine readable formats due to the wide use of electronic text, and computing resources are cheap, making it possible to process large amounts of data. Unfortunately, most unsupervised systems do not yield results that could outperform supervised learning, or look promising in the future prospects of being able to do so.

Yet another group of research was based on knowledge based methods, which tried to utilize the human knowledge encoded in various linguistic resources, such as lexical databases or dictionaries. This latter group face the problem that such sources are

generally very sparse, and do not yield enough information that can be used in typical WSD.

The approach of learning generic word sense classes that is presented in this thesis attempts to handle the issue of data sparseness, by making it possible to reuse the available amounts of training data; this is done by generalizing the patterns found within the data over different word senses. This thesis does not claim that this approach is a final solution for knowledge acquisition bottleneck; rather, it is presented as a way of maximizing the use of whatever the available amount of training data, assuming beforehand that available training data is sparse. Although the quality of labeled data created by professional linguists cannot be underestimated, the high cost of their availability and their genre- and time/corpus- dependent nature justifies, in our opinion, the investigations of strategies that address the question of how much of word sense knowledge can be generalized.

At least a part of this 'knowledge' that can be generalized, this thesis shows, is not necessarily semantic —as was the case with aforementioned manually-encoded knowledge based systems— but *contextual usage* oriented. This work suggests that the word senses that have similar usage patterns may be used as substitute examples for each other, with some constraints applied on the training process.

This approach is different from the tradition of machine learning, because it tries to refine the *classes* so they are easier to learn, rather than the optimizing either classifier or feature representation. Essentially, both class and feature optimization can be thought of as model optimization methods. Still, optimizing classes can be useful in cases where the problem is such that

- there is no established set of classes that one must use, or there is an easy way to convert any set of generic classes into the desired (established, possibly non-generic) set of classes
- there is no obvious theoretical way to determine which classes are the best for the task

The way this thesis sees the problem of WSD, supported by theoretical work of

many others that was mentioned in early chapters, matches both criteria; in addition, there are good reasons to believe that features and classifiers used in WSD are reasonably well-performing already — there has not been any significant change in different features used, or different classifier systems introduced, during last few years of WSD research. This makes 'class refinement' an interesting research problem for WSD.

8.1 Our Contribution

The core contribution of this work is to demonstrate the utility of generalizing sense knowledge for the end task of fine-grained word sense disambiguation. There have been attempts that utilized the 'common' knowledge encoded within thesauri, dictionaries, and lexical databases such as WORDNET; however they either were limited in scale —picking a few selected word examples and a matching set of homograph level senses— (Yarowsky, 1992), or just resorted to substitute examples from hierarchical neighbors such as monosemous synonyms. To the best of our knowledge, ours is the first work that proposed a comprehensive framework of leaning generic word sense classes for fine grained *unrestricted* WSD, and also introduced the use of semantic similarity in order to constrain the machine learning algorithm used.

In addition we argued that, rather than founding the sense classes on a taxonomy that represents human-understandable semantics of the concept, a more practical approach would be to base the common sense classes on linguistic usage patterns. Linguists have been arguing for similar ideas, but most of these work (Levin, 1993; Wierzbicka, 1984; Wierzbicka, 1996) is outside computational lexical semantics, let alone contemporary WSD. Sense classification that is based on lexical and syntactic features has been proposed in the context of unsupervised word sense disambiguation (Pedersen and Kulkarni, 2005), and this work is concerned on sense discrimination per word rather than generalizing usage patterns over different words. It is the generalization that makes it possible to pick training examples from different words. We believe that our approach of creating sense classes of different words, based on their contextual patterns, and using them in supervised fine grained WSD as an alternative

to human-created classes, is novel in the context of word sense disambiguation.

8.2 Further Work

Some questions we think are worth further research are discussed here, though they are outside the scope of this work. Some were not feasible to implement due to technical and time constraints.

8.2.1 Issue of Noise

One problem regarding feature-based clustering is that it is susceptible to errors and noise, even after measures such as singular value decomposition is employed. A major reason for this is that the sense-labeled corpus is small from any standards, and does not represent all senses of the words and their representative features.

One way to fix this problem is to employ very large quantities of unlabeled data during the calculation of the singular value decomposition of the feature vectors; this is possible as SVD does not care about labels of instances. Once the SVD vectors are calculated, the unlabeled instances can be discarded. In our case, we were limited by the hardware constraints and could not increase the amount of instances, as this would mean an increase in dimension of data vectors. It would be worthwhile to investigate the effect of a much 'cleaner' singular value decomposition on the resulting quality of classes.

8.2.2 Definitive Senses and Semantics

On several occasions, the words *dog* and *cat* were taken as examples to show that the dominant sense has the semantics of PET instead of ANIMAL. These senses were compared to related synonyms such as *domestic dog*. Nevertheless, the ANIMAL semantics is there, associated with the word, and is probably the next major meaning in terms of frequency. Of course, WORDNET does not distinguish this difference even in fine grained senses, but an interesting question to ask is: is it possible to reliably learn these differences without manual intervention?

If it is, the use of such a system is readily visible even without a way to attach semantics to the clusters thus formed (which would possibly require manual labor). Lexicographers have been using similar approaches (Tugwell and Kilgarriff, 2000) for a long time, in order to identify what senses are present in a given corpus. They then proceeded with manual assignments; in our case, the resulting clusters would hint the system which senses of a word are worth learning separately and should not be confused with others. This will greatly help reduce conflicts during the classifier process.

Again, the key for such an enterprise might finally lie on the availability of vast quantities of unlabeled data. This is an open issue, and is worth a further thought.

8.2.3 Automatically Labeling Generic Sense Classes

Another interesting question is related to a problem that was skipped above: applications which will actually require semantic labels for the learned classes. As we showed and discussed in this work, this is not an essential requirement for fine grained word sense disambiguation. But in other applications, labels coming with the classes might be of actual use. For instance in lexicography, as was mentioned earlier (Tugwell and Kilgarriff, 2000; Kilgarriff, 1997c), the clusters formed could possibly be manually inspected and labels assigned. In applications such as information extraction, question answering, and information retrieval, where named entity recognition is shown to have utilities, it may be possible to help the process if we can identify generic semantic classes as entities that come with meaning labels attached.

In cases like this, WORDNET lexicographer files could possibly have better utilities than our automatically generated clusters. It *may be* theoretically possible to attach generic semantics to automatically formed clusters. To take an example from section 5.1.2, although the WORDNET hypernym of *fishing rod* is *rod*, there is a separate entry for *sports equipment* in WORDNET. If we can identify semantics of the concepts in such a way that we can group *fishing rod*, *football*, *badminton racquet* as belonging to the same class, and then identify *sports equipment* as the best generic 'representative' of this class, it provides an adequate method of automatic sense labeling, at least for some purposes. Attaching a label to the group can be reduced to identifying the representative sense,

which is the *semantic* parent of the group rather than taxonomical.

It can be expected that this approach will provide an easy way of 'labeling' the clusters we find automatically, and it would be interesting to research how this can be made practically possible.

References

Agirre, Eneko and Oier Lopez de Lacalle. 2003. Clustering WordNet word senses. In *Proceedings of the Conference on Recent Advances on Natural Language Processing (RANLP '03)*, Borovets, Bulgary.

Agirre, Eneko and Oier Lopez de Lacalle Lekuona. 2004. Publicly available topic signatures for all WordNet nominal senses. In *Proceedings of the 4rd International Conference on Languages Resources and Evaluations (LREC)*, Lisbon, Portugal.

Agirre, Eneko and David Martinez. 2004. Unsupervised wsd based on automatically retrieved examples: The importance of bias. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain.

Atkins, Sue. 1992–93. Tools for corpus-aided lexicography: the hector project. *Acta Linguistica Hungarica*, 41:5–72.

Aurenhammer, Franz. 1991. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, September.

Banerjee, Satanjeev and Ted Pedersen. 2002. An adapted Lesk algorithm for word sense disambiguation using WORDNET . In *Proceeding of the Fourth International Conference on Computational Linguistics and Intelligent Text Processing (CICLING-02)*, Mexico City.

Bar-Hillel, Yehoshua. 1964. *Language and information: Selected essays on their theory and application*. Addison-Wesley, Reading, MA.

Bar-Hillel, Yehoshua. 1970. *Aspects of language: Essays in philosophy of language, linguistic philosophy, and methodology of linguistics*. Magnes Press, Jerusalem and North-Holland Press, Amsterdam.

Basili, Robert, Michelangelo Della Rocca, and Maria Teresa Pazienza. 1997. Towards a bootstrapping framework for corpus semantic tagging. In *Proceedings of the SIGLEX Workshop Tagging Text with Lexical Semantics: What, why and how?*, Washington, D.C., April. ANLP.

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Carpuat, Marine and Dekai Wu. 2005. Word sense disambiguation vs. statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 387–394, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Chan, Yee Seng and Hwee Tou Ng. 2005. Word sense disambiguation with distribution estimation. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 1010–1015, Edinburgh, Scotland, UK.

Chan, Yee Seng and Hwee Tou Ng. 2006. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of joint conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006)*, Sydney, Australia, July.

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

Choueka, Y. and S. Lusignan. 1985. Disambiguation by short context. *Computers and Humanities*, 19:147–157.

Ciaramita, Massimiliano and Mark Johnson. 2000. Explaining away ambiguity: learning verb selectional preference with bayesian networks. In *Proceedings of the 18th conference on Computational linguistics*, pages 187–193, Morristown, NJ, USA. Association for Computational Linguistics.

Ciaramita, Massimiliano and Mark Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Cortes, Corinna and Vladimir Vapnik. 1995. Support vector network. *Macine Learning*, 20(3):273–297, September.

Cottrell, Garrison W. 1989. *A connectionist approach to word sense disambiguation*. Pitman, London and Morgan Kaufmann, Los Altos, CA.

Cover, T.M. and P. E. Hart. 1967. Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, 13(1):21 – 27.

Crestan, Eric. 2004. Contextual semantics for WSD. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 101–104, Barcelona, Spain, July. Association for Computational Linguistics.

Crestan, Eric, M. El-Bze, and C. De Loupy. 2001. Improving WSD with multi-level view of context monitored by similarity measure. In *Proceeding of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*, Toulouse, France, July.

Curran, James. 2005. Supersense tagging of unknown nouns using semantic similarity. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 26–33, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Curtis, Jon, David Baxter, and John Cabral. 2006. On the application of the cyc ontology to word sense disambiguation. In *Proceedings of the Nineteenth International FLAIRS Conference*, pages 652–657, Melbourne Beach, FL, May.

Cutting, Douglass R., David R. Karger, Jan O. Pedersen, and John W. Tukey. 1992. Scatter/gather: a cluster-based approach to browsing large document collections. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–329, New York, NY, USA. ACM Press.

Daelemans, Walter. 1999. Introduction to the special issue on memory-based language processing. *Journal of Experimental and Theoretical Artificial Intelligence*, 11(3):287 – 296.

Daelemans, Walter, Antal van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3):11–41.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2004. Timbl: Tilburg memory based learner, version 5.1, reference guide. Technical report, ILK Technical Report Series 04-02, Available from http://ilk.uvt.nl/downloads/pub/papers/ilk.0310.pdf.

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1997. Similarity-based methods for word sense disambiguation. In *Proceedings of the 35th annual meeting on Association for Computational Linguistics*, pages 56–63, Morristown, NJ, USA. Association for Computational Linguistics.

Dagan, Ido, Shaul Marcus, and Shaul Markovitch. 1993. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31st conference on Association for Computational Linguistics*, pages 164–171. Association for Computational Linguistics.

Dagan, Ido, Fernando Pereira, and Lillian Lee. 1994. Similarity-based estimation of word cooccurrence probabilities. In *Proceedings of the 32nd conference on Association for Computational Linguistics*, pages 272–278. Association for Computational Linguistics.

Decadt, Bart, Véronique Hoste, Walter Daelemans, and Antal Van den Bosch. 2004. Gambl, genetic algorithm optimization of memory-based wsd. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 108–112, Barcelona, Spain, July. Association for Computational Linguistics.

Deerwester, Scott C., Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and

Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Edmonds, Phil and Scott Cotton. 2001. Senseval-2: Overview. In *Proc. of the Second Intl. Workshop on Evaluating Word Sense Disambiguation Systems (Senseval-2)*, Toulouse, France, July.

Fellbaum, Christiane. 1998a. WORDNET - *An Electronic Lexical Database*. The MIT Press, Cambridge, MA.

Fellbaum, Christiane, 1998b. WORDNET - *An Electronic Lexical Database*, chapter Semantic Network of English Verbs, pages 69–104. The MIT Press, Cambridge, MA.

Férnandez-Amorós, David. 2004. Wsd based on mutual information and syntactic patterns. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 117–120, Barcelona, Spain, July. Association for Computational Linguistics.

Francis, W. Nelson and Henry Kucera. 1982. *Frequency Analysis of English Usage*. Houghton Miffli.

Friedman, Jerome H. 1996. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University.

Gale, William, Kenneth Church, and David Yarowsky. 1992. Using bilingual materials to develop word sense disambiguation methods. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 101–112.

Gliozzo, Alfio, Carlo Strapparava, , and Ido Dagan. 2004. Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. *Computer Speech and Language*, 18:275–299.

Golub, G. H. and C. Reinsch. 1970. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403 – 420, April.

Graff, David. 2002. The AQUAINT corpus of English news text. Linguistic Data Consortium (LDC), LDC2002T31.

Gross, D., U. Fischer, and G. A. Miller. 1989. The organization of adjectival meanings. *Journal of Memory and Language*, 28:92–106.

Guan, Yu, Ali A. Ghorbani, and Nabil Belacel. 2004. K-means+: An autonomous clustering algorithm. Technical Report TR04-164, University of New Brunswick.

Hirst, Graeme and David St-Onge, 1998. *WordNet: An electronic lexical database*, chapter Lexical Chains as representations of context for the detection and correction of malapropisms, pages 305–332. MIT Press.

Hoste, Véronique, Walter Daelemans, Iris Hendrickx, and Antal van den Bosch. 2002. Evaluating the results of a memory-based word-expert approach to unrestricted word sense disambiguation. In *Proceedings of the Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 95–101, Philadelphia, PA, USA.

Hoste, Véronique, Anne Kool, and Walter Daelmans. 2001. Classifier optimization and combination in english all words task. In *Proceeding of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 83–86, Toulouse, France, July.

Ide, Nancy and Jean Véronis. 1998. Word sense disambiguation: The state of the art. *Computational Linguistics*, 241:1 – 40.

Jain, A. K. and R. C. Dubes. 1988. *Algorithms for Clustering Data*. Prentice Hall.

Jastrzembski, James E. 1981. Multiple meanings, number of related meanings, frequency of occurrence, and the lexicon. *Cognitive Psychology*, 13:278–305.

Jastrzembski, James E. and Robert F. Stanners. 1975. Multiple word meanings and lexical search speed. *Journal of Verbal Learning and Verbal Behavior*, 14:534–537.

Jiang, Jay and David Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, Taiwan.

Katz, Jerrold J. and Jerry A. Fodor. 1963. The structure of a semantic theory. *Language*, 39(2):170–210, April - June.

Kilgarriff, Adam. 1993. Dictionary word sense distinctions: An enquiry into their nature. *Computers and the Humanities*, 26(1–2):365–387.

Kilgarriff, Adam. 1997a. Foreground and background lexicons and word sense disambiguation for information extraction. Technical report, ITRI 97–08. Also published in Proc. Workshop on Lexicon Driven Information Extraction, Frascati, Italy.

Kilgarriff, Adam. 1997b. I don't believe in word senses. *Computers and the Humanities*, 31(2):91 – 113, March.

Kilgarriff, Adam. 1997c. What is word sense disambiguation good for? Technical report, ITRI-97-08. Also published in Proc. NLP Pacific Rim Symposium '97, Phuket, Thailand.

Kilgarriff, Adam. 2000. Framework and results for english SENSEVAL. Technical Report ITRI-00-20, Information Technology Research Institute, University of Brighton. Also published in Computers and the Humanities 34 (1–2), Special Issue on SENSEVAL, pp 15–48.

Kilgarriff, Adam and David Tugwell. 2001a. WASP-Bench: an MT lexicographers' workstation supporting state-of-the-art lexical disambiguation. In *Proc. of MT Summit VII*, pages 187–190, Santiago de Compostela.

Kilgarriff, Adam and David Tugwell. 2001b. WORD SKETCH: Extraction and display of significant collocations for lexicography. In *Proc. Collocations workshop, ACL 2001*, pages 32–38, Toulouse, France.

Koeling, Rob, Diana McCarthy, and John Carroll. 2005. Domain-specific sense distributions and predominant sense acquisition. In *Proceedings of the Human Language*

Technology Conference and Conference on Empirical Methods in Natural Language Processing. HLT/EMNLP 2005, pages 419–426, Vancouver, B.C., Canada, October.

Korhonen, Anna. 2002. Assigning verbs to semantic classes via wordnet. In *Proceedings of the COLING Workshop on Building and Using Semantic Networks*.

Krovets, Robert and Bruce Croft. 1992. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2):115–141.

Kullback, Solomon and R. A. Leibler. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79 – 86, March.

Landes, Shari, Claudia Leacock, and Randee I. Tengi, 1998. WORDNET - *An Electronic Lexical Database*, chapter Building Semantic Concordances, pages 199–216. The MIT Press, Cambridge, MA.

Leacock, C., G. Towell, and E. Voorhees. 1993. Corpus-based statistical sense resolution. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 260–265, March.

Leacock, Claudia and Martin Chodorow, 1998. WORDNET - *An Electronic Lexical Database*, chapter Combining local context and WordNet similarity for word sense identification, pages 265–283. The MIT Press, Cambridge, MA.

Leacock, Claudia, George A. Miller, and Martin Chodorow. 1998. Using corpus statistics and WORDNET relations for sense identification. *Computational Linguistics*, 24(1):147–165.

Lee, Yoong Keok and Hwee Tou Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, pages 41–48, Philadelphia, Pennsylvania, USA.

Lee, Yoong Keok, Hwee Tou Ng, and Tee Kiah Chia. 2004. Supervised word sense disambiguation with support vector machines and multiple knowledge sources. In

Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 137–140, Barcelona, Spain, July. Association for Computational Linguistics.

Lenat, Doug, George Miller, and Toshio Yokoi. 1995. CYC, WordNet, and EDR: critiques and responses. *Communications of the ACM*, 38(11):45–48.

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

Lesk, Michael. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation (SIGDOC '86)*, pages 24–26, New York, NY, USA. ACM Press.

Levin, Beth. 1993. *English Verb Classes And Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago and London.

Lin, Dekang. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of Annual Meeting of Association for Computational Linguistics*, Madrid, Spain, July.

Litkowski, Kenneth C. 2000. SENSEVAL: The CL Research experience. *Computers and the Humanities*, 34(1–2):153–158.

Litkowski, Kenneth. C. 2001. Use of machine readable dictionaries for word-sense disambiguation in SENSEVAL-2. In *Proceedings of Association for Computational Linguistics Special Interest Group on the Lexicon Workshop*, Toulouse, France.

Littlestone, N and M.K. Warmuth. 1994. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, February.

Magnini, Bernardo and Gabriela Cavaglià. 2000. Integrating subject field codes into WORDNET . In M. Gavrilidou, G. Crayannis, S. Markantonatu, S. Piperidis, and G. Stainhaouer, editors, *Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation*, pages 1413–1418, Athens, Greece, May–June.

Marcus, Mitchell, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn treebank: Annotating predicate argument structure. In *Proceedings of ARPA Human Language Technology Workshop*.

Màrquez, Lluís. 2000. Machine learning and natural language processing. Technical Report LSI-00-45-R, Departament de Llenguatges i Sistemes Informàtics (LSI), Universitat Politecnica de Catalunya (UPC), Barcelona, Spain.

Màrquez, Lluís. 2004. Senseval-3 panel on planning Senseval-4. Presentation, July. http://www.cs.unt.edu/∼rada/senseval/senseval3/panels/Marquez.pdf.

Martinez, David and Eneko Agirre. 2000. One sense per collocation and genre/topic variations. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, Hong Kong.

Masterman, Margaret. 1961. Semantic message detection for machine translation using an interlingua. In *Proceedings of the International Conference on Machine Translation*, pages 438–475.

McCarthy, Diana and John Carroll. 2003. Disambiguating nouns, verbs and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, 29(4):639–654.

Mihalcea, Rada. 2002. Word sense disambiguation using pattern learning and automatic feature selection. *Journal of Natural Language and Engineering (JNLE)*, 1(1):1–15, December.

Mihalcea, Rada and Tim Chklovski. 2002. Building a sense tagged corpus with open mind word expert. In *Proceedings of the ACL 2002 Workshop on 'Word Sense Disambiguation: Recent Successes and Future Directions'*, Philadelphia, July.

Mihalcea, Rada and Timothy Chklovski. 2003. Open mind word expert: Creating large annotated data collections with web users' help. In *Proceedings of the EACL 2003 Workshop on Linguistically Annotated Corpora (LINC 2003)*, Budapest, April.

Mihalcea, Rada, Timothy Chklovski, and Adam Kilgarriff. 2004. The Senseval-3 english lexical sample task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Barcelona, Spain, July. Association for Computational Linguistics.

Mihalcea, Rada and Ehsanul Faruque. 2004. SenseLearner: Minimally supervised word sense disambiguation for all words in open text. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 155–158, Barcelona, Spain, July. Association for Computational Linguistics.

Mihalcea, Rada, Adam Kilgariff, Lluís Màrquez, Ido Dagan, Martha Palmer, and Philip Resnik. 2004. Senseval-3 panel on planning Senseval-4. Meeting Minutes, July. http://www.cs.unt.edu/∼rada/senseval/senseval3/panels/ minutes-senseval4-panel.txt.

Mihalcea, Rada and Dan I. Moldovan. 2000. An iterative approach to word sense disambiguation. In *Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference*, pages 219–223. AAAI Press.

Miller, George A, 1998. WORDNET - *An Electronic Lexical Database*, chapter Nouns in WordNet, pages 23 – 46. The MIT Press, Cambridge, MA.

Miller, George A. and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processe*, 6(1):1–28.

Mitchell, Tom. 1997. *Machine Learning*. McGraw Hill.

Ng, Hwee Tou. 1997. Getting serious about word sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, pages 1–7, Washington, D.C., USA.

Ng, Hwee Tou and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th*

Annual Meeting of the Association for Computational Linguistics, pages 40–47, Santa Cruz, California, USA, June.

Ng, Hwee Tou, Chung Yong Lim, and Shou King Foo. 1999. A case study on inter-annotator agreement for word sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Standardizing Lexical Resources (SIGLEX99)*, pages 9–13, College Park, Maryland, USA.

O'Hara, Tom, Rebecca Bruce, Jeff Donner, and Janyce Wiebe. 2004. Class-based collocations for word sense disambiguation. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 199–202, Barcelona, Spain, July. Association for Computational Linguistics.

Olsen, Mari Broman, Bonnie J. Dorr, and David J. Clark. 1997. Using wordnet to posit hierarchical structure in Levin's verb classes. Technical report, University of Maryland at College Park, College Park, MD, USA.

Patwardhan, Siddharth, Satanjeev Banerjee, and Ted Pedersen. 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, February.

Pedersen, Ted and A. Kulkarni. 2005. Identifying similar words and contexts in natural language with senseclusters. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, Pittsburgh, PA, July.

Pedersen, Ted and Anagha Kulkarni. 2006. Automatic cluster stopping with criterion functions and the gap statistic. In *Proceedings of the Demonstration Session of the Human Language Technology Conference and the Sixth Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, New York City, June.

Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. 2004. Word-

Net::Similarity - measuring the relatedness of concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, San Jose, CA, July.

Pereira, Fernando and Naftali Tishby. 1992. Distributional similarity, phase transitions and hierarchical clustering. Goldman, R., editor, Fall Symposium on Probability and Natural Language. AAAI. Cambridge, Mass.

Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of english words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 183–190, Morristown, NJ, USA. Association for Computational Linguistics.

Pickett, Joseph P. et al., editors. 2000. *The American Heritage Dictionary of the English Language*. Houghton Mifflin, Boston, fourth edition.

Procter, Paul. 1978. *Longman Dictionary of Contemporary English*. Longman Group, Harlow, Essex, England.

Pustejovsky, James. 1995. *The Generative Lexicon*. MIT Press, Cambridge, Massachusetts, USA.

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

Resnik, Philip. 1993. *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

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

Resnik, Philip. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61:127–159, November.

Resnik, Philip. 1997. Selectional preference and sense disambiguation. In *Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How?*, Washington, April.

Resnik, Philip and David Yarowsky. 1997. A perspective on word sense disambiguation methods and their evaluation. In Marc Light, editor, *Proceedings of Workshop of SIGLEX (Lexicon Special Interest Group) of the ACL on Tagging Text with Lexical Semantics: Why, What and How?*, pages 79–86, Washington, April.

Rubenstein, Herbert and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Sanderson, Mark. 1994. Word sense disambiguation and information retrieval. In *Proceedings of 17th ACM Specical Interest Group on Information retrieval (SIGIR)*, pages 142–151.

Schank, Roger C. 1973. The fourteen primitive actions and their inferences. Technical report, Stanford, CA, USA.

Seo, Hee-Cheol, Hae-Chang Rim, and Soo-Hong Kim. 2004. KUNLP system in Senseval-3. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 222–225, Barcelona, Spain, July. Association for Computational Linguistics.

Sleator, Daniel and Davy Temperley. 1991. Parsing English with a link grammar. Technical report, Carnegie Mellon University Computer Science CMU-CS-91-196, October.

Snyder, Benjamin and Martha Palmer. 2004. The English all-words task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain, July. Association for Computational Linguistics.

Stevenson, Suzanne and Paola Merlo. 2000. Automatic lexical acquisition based on sta-

tistical distributions. In *Proceedings of the 17th conference on Computational linguistics*, pages 815–821, Morristown, NJ, USA. Association for Computational Linguistics.

Strapparava, Carlo, Alfio Gliozzo, and Claudiu Giuliano. 2004. Pattern abstraction and term similarity for word sense disambiguation: IRST at Senseval-3. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 229–234, Barcelona, Spain, July. Association for Computational Linguistics.

Tengi, Randee I., 1998. WORDNET - *An Electronic Lexical Database*, chapter Design and Implementation of the WordNet Lexical Database and Searching Software, pages 105–127. The MIT Press, Cambridge, MA.

Tibshirani, R., G. Walther, , and T. Hastie. 2001. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistics Society (Series B)*, page 411423.

Tugwell, David and Adam Kilgarriff. 2000. Harnessing the lexicographer in the quest for accurate word sense disambiguation. In *Proceedings of 3rd International Workshop on Text, Speech, Dialogue (TSD 2000)*, pages 9–14, Brno, Czech Republic. Springer Verlag Lecture Notes in Artificial Intelligence.

Vapnik, Vladimir. 1998. *Statistical Learning Theory*. Wiley-Interscience, Ney York, NY, September.

Vapnik, Vladimir. 1999. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Véronis, Jean. 1998. A study of polysemy judgements and inter-annotator agreement. In *Programme and advanced papers of the Senseval workshop*, pages 2–4, Herstmonceux Castle, England, September.

Villarejo, Luís, Lluis Màrquez, Eneko Agirre, David Martínez, Bernardo Magnini, Carlo Strapparava, Diana McCarthy, Andrés Montoyo, and Armando Suárez. 2004. The

"Meaning" system on the English all-words task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 253–256, Barcelona, Spain, July. Association for Computational Linguistics.

Wagstaff, Kiri, Claire Cardie, Seth Rogers, and Stefan Schroedl. 2001. Constrained k-means clustering with background knowledge. In *Proceedings of 18th International Conference on Machine Learning (ICML-01)*, pages 577–584.

Wall, Michael E., Andreas Rechtsteiner, and Luis M. Rocha, 2003. *A Practical Approach to Microarray Data Analysis*, chapter Singular value decomposition and principal component analysis, pages 91–109. Kluwer, Norwell, MA.

Weaver, Warren. 1949. Translation. *Momeographed*, pages 15–23. Repr. in: Locke, W.N. and Booth, A.D. (eds.) Machine translation of languages: fourteen essays.

Wierzbicka, Anna. 1984. "Apples" are not a "kind of fruit": The semantics of human categorization. *American Ethnologist*, 11(2):313–328, May.

Wierzbicka, Anna, 1996. *Semantics: primes and universals*, chapter Semantics and Ethnobiology, pages 351–376. Oxford University Press.

Wilks, Yorick. 1968. *Argumment and Proof*. Ph.D. thesis, Cambridge University.

Wilks, Yorick. 1975. Primitives and words. In *Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, pages 38–41. Association for Computational Linguistics.

Wilks, Yorick. 1997. Senses and texts. *Computers and the Humanities*, 31(2):77–90, March.

Wilks, Yorick. 1998. Is word-sense disambiguation just one more NLP task? In *Proceedings of SENSEVAL Conference, Herstmonceaux, Sussex*. Also appears as Technical Report CS-98-12, Department of Computer Science, University of Sheffield.

Wilks, Yorick and Mark Stevenson. 1996. The grammar of sense: Is word-sense tagging much more than part-of-speech tagging? Sheffield Department of Computer Science, Research Memoranda, CS-96-05.

Wilks, Yorick and Mark Stevenson. 1998. The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Natural Language Engineering*, 4(2):135–143.

Witten, Ian H., Alistair Moffat, and Timothy C. Bell. 1999. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishing, San Francisco.

Yarowsky, David. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of COLING-92*, pages 454–460, Nantes, France, July.

Yarowsky, David. 1993. One sense per collocation. In *Proceedings of ARPA Human Language Technology Workshop*, pages 266–271, Princeton.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, MA.

Zhao, Ying and George Karypis. 2005. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168.

Zipf, G. K. 1945. The meaning-frequency relationship of words. In *The Journal of General Psychology*, volume 33. pages 251–256.

Appendix A
Other Clustering Methods

This section describes the detailed experimental results for the sense clustering systems that were rejected due to their undesirable properties, most importantly the poor performance on development data, along with relevant observations.

Data for this clustering algorithms come from the same vector models we described earlier in section 6.2. For both nouns and verbs, the coordinates were created using averaging of all instances within a sense. Dimension was also reduced using singular value decomposition.

Before settling for k-means+ algorithm described in section 6.3, we experimented with several standard clustering algorithms. Reported here are the results for two hierarchical clustering algorithms. The two clustering attempts are based on agglomerative and divisive clustering strategies, which either repeatedly merge or divide clusters until the required number of clusters is obtained. In addition to this, a method for automatically acquiring the number of clusters was also evaluated.

A.1 Clustering Schemes

There is a diverse array of clustering schemes in literature, each with its own advantages and drawbacks. However, most of these schemes differ from each other only in minor implementation detail; basic intuitions behind the schemes remain more or less the same. Because of this reason, and due to the practical constraints on resources,

systems tested here are only a fairly representative subset of the available clustering algorithms, rather than an exhaustive collection. As mentioned in section 6.7.3, the idea was not to conduct an exhaustive search in the first place, but to analyze the basic necessary features for clusters. These experiments are not conclusive on the matter which clustering scheme yields the best sense classes. This is an avenue for future research.

The clustering schemes discussed in this chapter are implemented using CLUTO clustering toolkit (Zhao and Karypis, 2005).

A.1.1 Agglomerative Clustering

One intuitive way to clump a large number of points to a smaller number of clusters is to keep merging points into clusters, and smaller clusters into larger ones. This 'bottom-up approach' of clustering can be proceeded until one ends up with a single 'root' cluster. In practice, we stop when the desired number of clusters is obtained. Which two clusters to merge at each step is determined by the particular clustering criterion function in use (discussed below).

A.1.2 Divisive Clustering

The converse of agglomerative clustering is to start with a single universal cluster that includes all points, and then keep dividing it (and the resulting clusters) until the desired level of division is achieved. The implementation can be different in finer points such as the criterion function used in determining which cluster to select for splitting. CLUTO adopts an approach of using k-NN algorithm to split the selected cluster in to two. The cluster which gives the best overall quality of the system upon split (depending on the criterion function in use) is selected as the candidate for splitting.

A.1.3 Cluster Criterion Functions

Different heuristics can be used in determining how to proceed in each step of clustering. Some of these do not depend on the actual clustering algorithm in use, but are defined on the clusters themselves, as a measure of 'quality' of the resulting clusters.

In case of agglomerative clustering, some criteria can be based on obvious heuristics. For instance single-linkage criterion merges the two clusters considering the maximum pairwise similarity (minimum pairwise distance) between two clusters, among all permutations of pairs one can pick from two clusters. The two clusters that have the maximum similarity are merged together. Complete linkage decides on the maximum pairwise distance, and merges the two clusters that have the smallest distances between their furthest-apart points. UPGMA (Jain and Dubes, 1988), also known as average linkage or group average, selects as merger candidates the two clusters that have the largest average pairwise similarity between each other. Some measures are defined for the resulting set of clusters: for instance $\mathcal{I}_2$ (Zhao and Karypis, 2005; Cutting et al., 1992) is defined as $\sum_{r=1}^{k} n_r(\frac{1}{n_r^2}\sum_{v_i,v_j\in S_r}\cos(v_i,v_j))$ for k clusters $S_1, S_2 \ldots S_k$. Sense vectors within each cluster are denoted by v, and $\cos(v_i, v_j)$ is the familiar cosine similarity measure between two vectors. For the sake of brevity we do not discuss all criterion functions here; they are described in detail in (Zhao and Karypis, 2005).

A.2 Comparison

In this section, we will compare agglomerative clustering with divisive.

Figures A.2 and A.1 show the sizes and organization of the clusters created by agglomerative and divisive clustering, at 20 clusters for verbs and 30 for nouns. One immediately obvious result is that agglomerative clustering produces very uneven clustering results. This kind of behavior is the case in general when we use simple-linkage as criterion functions, but not so usual for UPGMA which was used as the clustering criterion function in this experiments; however in this case, UPGMA does not give an even distribution, although its performance is still better than simple linkage and complete linkage methods. As we see in section A.2.1, this results in large sense loss in the case of nouns.

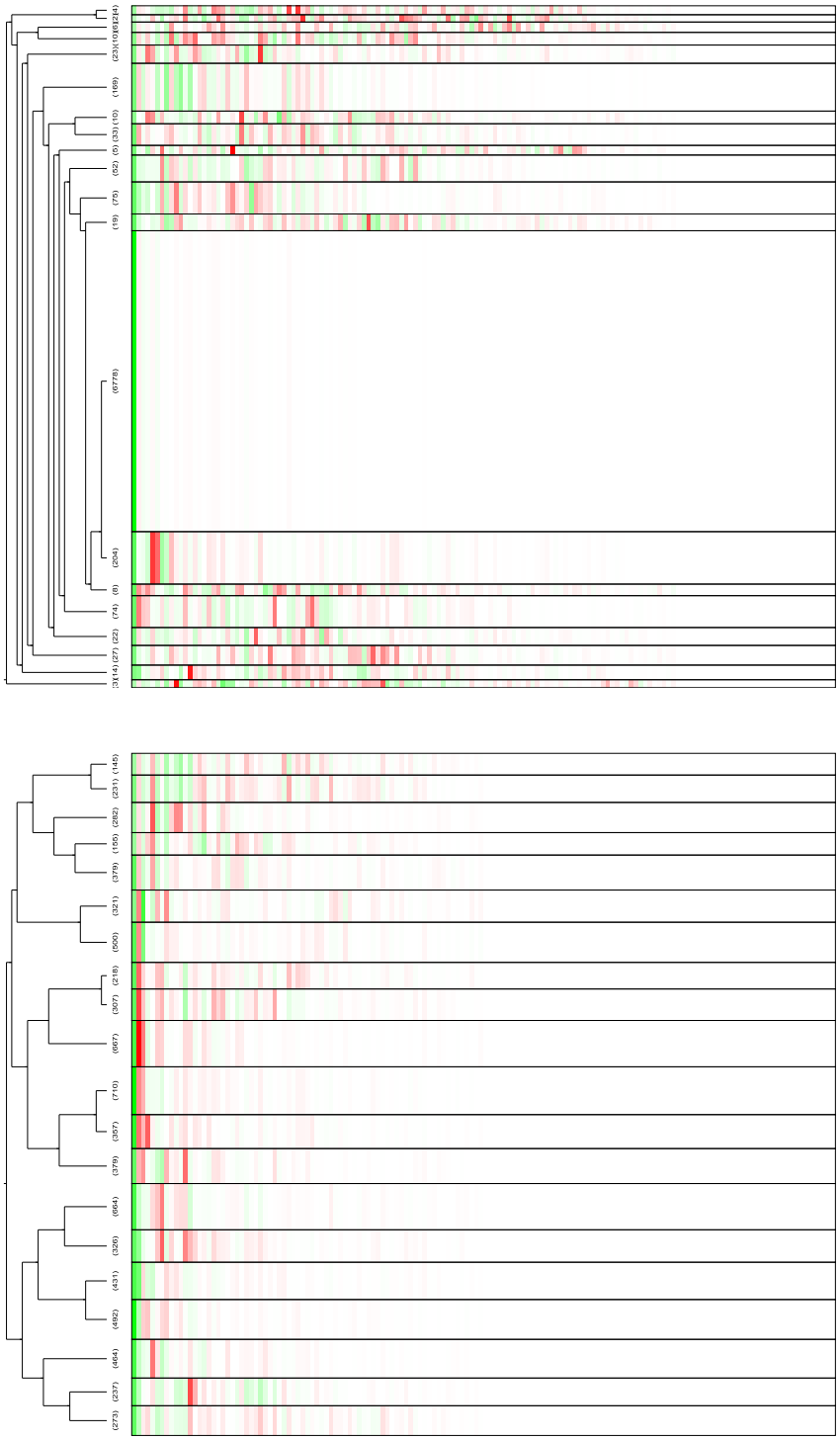

Figure A.1: Cluster distribution of verbs (part of speech feature, at 20 clusters) for agglomerative (above) and repeated bisection (below) methods. Numbers shown inside brackets are the number of senses in the cluster. Red and Green bars denote positive and negative values of feature vector (after SVD), and the color intensities denote the magnitude. The height of a cluster 'belt' is proportional to the number of points in the cluster; the figure shows that the agglomerative clustering has very uneven distribution, and a larger hierarchical depth.

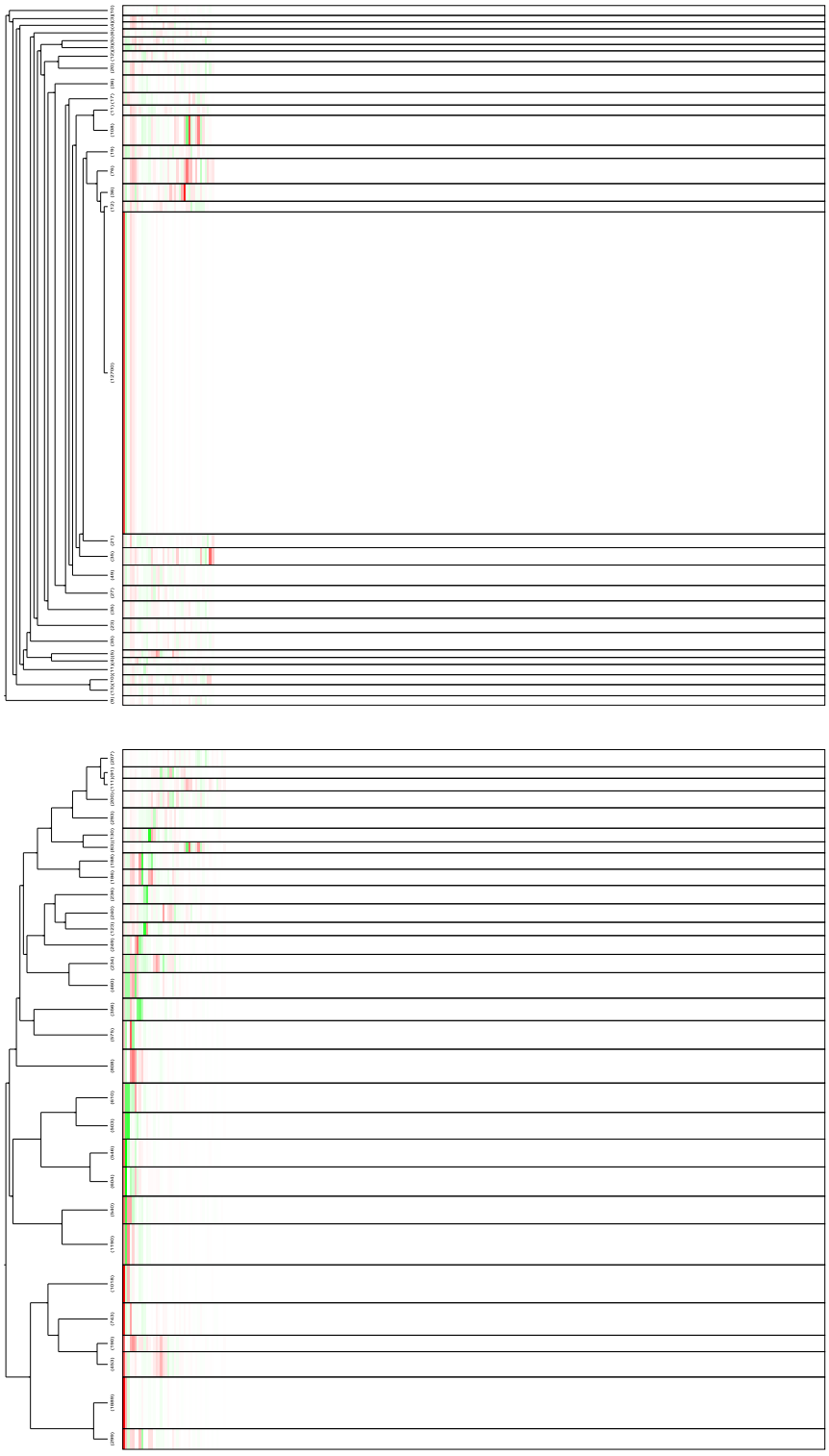

Figure A.2: Cluster distribution of nouns (local context feature, at 30 clusters) for agglomerative (above) and repeated bisection (below) methods. Numbers shown inside brackets are the number of senses in each cluster. As in the case for verbs, (figure A.1), agglomerative clustering results in badly distributed clusters, and a larger hierarchical depth.

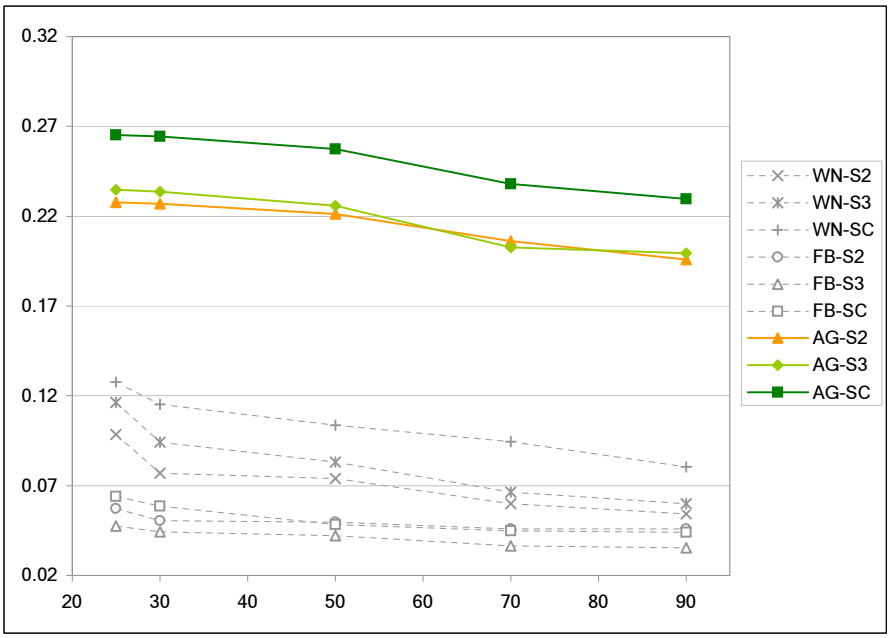

Figure A.3: Sense loss for agglomerative clustering for nouns. Shown in dotted lines are the sense loss graphs of WORDNET tree splits and feature-based modified k-NN clustering schemes (from figure 6.3). WN: WORDNET tree splits, FB: feature based modified k-NN, AG: agglomerative clustering.

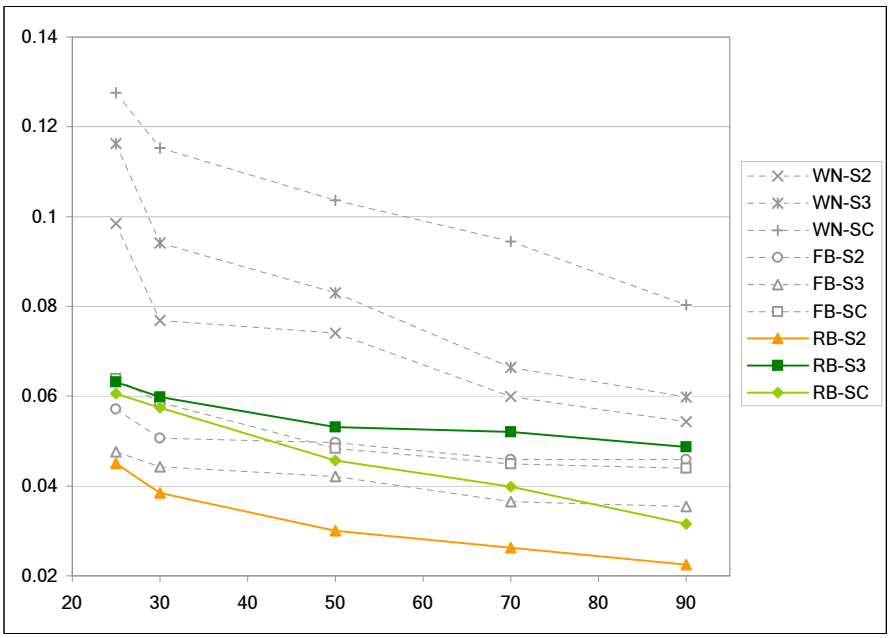

Figure A.4: Sense loss for repeated bisection clustering for nouns. RB: repeated bisection, rest of the details as per figure A.3.

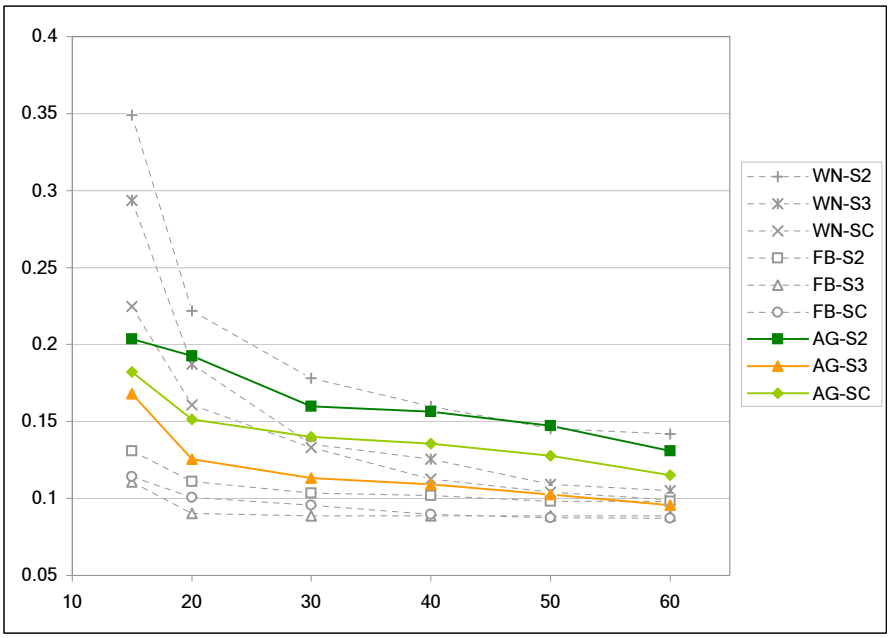

Figure A.5: Sense loss for agglomerative clustering for verbs. Shown in dotted lines are the sense loss graphs of WORDNET tree splits and feature-based modified k-NN clustering schemes (from figure 6.4). WN: WORDNET tree splits, FB: feature based modified k-NN, AG: agglomerative clustering.

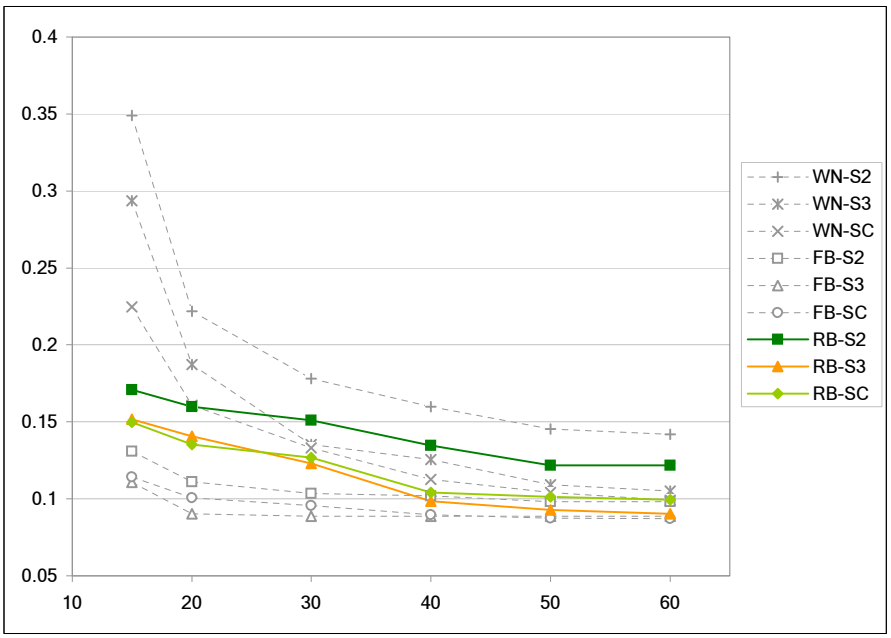

Figure A.6: Sense loss for repeated bisection clustering for verbs. RB: repeated bisection, rest of the details as per figure A.5.

A.2.1 Sense Loss

Figures A.3 and A.4 show the sense loss figures for agglomerative and repeated bisection clustering for nouns, together with sense loss figures of the clustering algorithms discussed in sections 6.3 and 6.4 (shown in dotted lines). Figures A.5 and A.6 show the same results for verbs.

What is evident from the figures is that agglomerative clustering scheme generally yields much worse sense loss figures, most of the time performing even worse than WORDNET hierarchy-segmenting based clusters. Repeated bisection, on the other hand, is comparatively better, and sometimes even outperforms our feature-based clustering in terms of sense loss.

Generally equal levels of performance of repeated bisection and our modified k-NN algorithm (section 6.3) can possibly be explained by the fact that the principal technique of our modified k-NN algorithm is reasonably close to repeated bisection than to agglomerative algorithm. However, it must be noted that the criteria for choosing the 'best' clustering scheme was not sense loss, but the performance of the WSD system on development data set. In this latter property, repeated bisection does not perform as well as the modified k-NN algorithm. This is partly due to the fact that repeated bisection does not allow rearrangements of senses between clusters once the clusters are determined. In this property, repeated bisection is more similar to the WORDNET hierarchy splitting algorithm (section 6.4). Although it has much better sense loss properties, the sense loss reduction itself does not guarantee a good classifier performance. For this reason, (which we discussed in detail in section 6.7.2) we can conclude that the modified k-NN algorithm we used for feature-based classes was the best for fine-grained WSD end-task, among the clustering algorithms that were tested.

A.2.2 SENSEVAL Performance

Tables A.1 and A.2 show the performance levels of sense class maps generated by agglomerative and divisive (repeated bisection) algorithms, in comparison with our modified k-NN algorithm on SENSEVAL tasks. In case of nouns, both agglomerative and re-

	SENSEVAL-2	SENSEVAL-3
Baseline	0.711	0.700
Agglomerative	0.713	0.701
Divisive	0.712	0.718
Modified k-NN	0.747	0.736

Table A.1: SENSEVAL performance of different clustering schemes: nouns

	SENSEVAL-2	SENSEVAL-3
Baseline	0.439	0.534
Agglomerative	0.437	0.549
Divisive	0.451	0.559
Modified k-NN	0.480	0.568

Table A.2: SENSEVAL performance of different clustering schemes: verbs

peated bisection clustering methods perform only marginally better than the baseline. However in the case of verbs there is some reasonable improvement in the repeated bisection method. What is interesting to observe is that the large sense loss in agglomerative clustering in case of nouns has not contributed much to make its noun performance much worse than that of divisive clustering. This is because divisive clustering, albeit with smaller sense loss, yield many answers that are wrong at class level. Recall that the sense loss measure does not say anything about the suitability of substitute senses, as it does not care about the relationship between senses of *different* words.

A.3 Automatically Deriving the Optimal Number of Classes

There has been several work in literature focusing on the problem of automatically deriving the number of classes. This is a model selection problem in way, as class systems at different numbers of classes can be thought of as various models that represent the actual underlying structure of a system. Similar to clustering criterion functions we discussed above, we can use various measures to determine where to stop clustering as well (Pedersen and Kulkarni, 2006).

Attempts to determine the number of sense classes automatically using these measures did not yield any productive outcome. The same clustering schemes we described above (agglomerative and repeated bisection) were used in a similar setting,

	Agglomerative		**RB**	
	noun	verb	noun	verb
PK2	3	1	1	1
Gap	5	3	2	2

Table A.3: Optimal numbers of clusters returned by automatic cluster stopping criterion functions. RB: Repeated bisection method.

while a stopping criterion was used to determine when to stop clustering. Our implementation used parts of SENSECLUSTERS (Pedersen and Kulkarni, 2005). The clustering stopping criterions tested are the Gap statistic (Tibshirani et al., 2001) and PK2 (Pedersen and Kulkarni, 2006).

The automatic stopping criterion did not yield any reasonable result for the number of clusters. The numbers of clusters returned as optimal are shown in table A.3 for both measures.

Obviously, these numbers are too coarse for our purpose, and hence not useful.

A.4 Summary

In this section we described several results that are related to the sense clustering experiments. On the development data set, our implementation of modified k-nearest neighbor algorithm performed better (in terms of end-task results) than the class maps generated from these algorithms. Similarly, automatic determination of clusters does not seem to yield any promising results in this particular experiment setting.