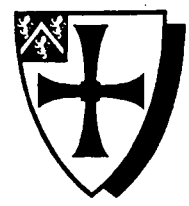
Durham - a word sense disambiguation system

Hawkins, Paul Martin

University of Durham

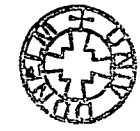


DURHAM - A Word Sense Disambiguation System

Paul Martin Hawkins

Laboratory for Natural Language Engineering

Department of Computer Science



September 1999

Submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

Abstract

DURHAM - A Word Sense Disambiguation System

Paul Martin Hawkins

Ever since the 1950's when Machine Translation first began to be developed, word sense disambiguation (WSD) has been considered a problem to developers. In more recent times, all NLP tasks which are sensitive to lexical semantics potentially benefit from WSD although to what extent is largely unknown.

The thesis presents a novel approach to the task of WSD on a large scale. In particular a novel knowledge source is presented named contextual information. This knowledge source adopts a sub-symbolic training mechanism to learn information from the context of a sentence which is able to aid disambiguation. The system also takes advantage of frequency information and these two knowledge sources are combined. The system is trained and tested on SEMCOR.

A novel disambiguation algorithm is also developed. The algorithm must tackle the problem of a large possible number of sense combinations in a sentence. The algorithm presented aims to make an appropriate choice between accuracy and efficiency. This is performed by directing the search at a word level.

The performance achieved on SEMCOR is reported and an analysis of the various components of the system is performed. The results achieved on this test data are pleasing, but are difficult to compare with most of the other work carried out in the field. For this reason the system took part in the SENSEVAL evaluation which provided an excellent opportunity to extensively compare WSD systems. SENSEVAL is a small scale WSD evaluation using the HECTOR lexicon. Despite this, few adaptations to the system were required. The performance of the system on the SENSEVAL task are reported and have also been presented in [Hawkins, 2000].

Acknowledgements

Firstly may I extend my thanks to the EPSRC and the Defence Evaluation Research Agency at Malvern for funding my work. From an academic view point, there are several people who I have much to thank for their input into this PhD. Roberto Garigliano's infinite depth of knowledge on such a wide range of topics is still quite baffling. His enthusiasm for the subject has rubbed off on many students, and I am fortunate to be one of them. Russell Collingham's input during my undergraduate study and first year of my PhD was very important. His influence was critical in my choice to stay at Durham and has helped revolutionise the image of computer scientists within Durham. Above all, my most sincere thanks go to Dave Nettleton who through out my entire PhD has always been there for me. I've always known he was only a knock on the door away and his support and encouragement as a supervisor and friend are something for which I am very grateful. It is slightly worrying that all of my supervisors have now left the University. I wish them every success at 3F, and as people are the most important asset for any company, I know they have a very prosperous future. Also, a big thanks goes to Lee Hollingdale my long suffering office mate. His great wit and knowledge of postscript has kept me sane through many of the difficult times.

Outside the University I would like to express my thanks to Mark Stevenson, George Paliouras and Jean Véronis for their contribution to some interesting email discussions. Also I would like to express a big thank you to Adam Kilgarriff, had he not taken up the challenge to organise SENSEVAL, chapter 7 would not have been possible!

From a personal perspective, I have a huge debt of thanks to my family. They have always been very supportive throughout my upbringing, and this work is as much a credit to them as it is to me. However, I am still worried about the time when they realise that being a doctor will not enable me to cure their aliments! Also my girlfriend, Chieko, has given me much love and patience during my final year. I am very grateful for all her clue word spotting and proof reading.

Declaration

The material contained within this thesis has not previously been submitted for a degree at the University of Durham or any other university. The research reported within this thesis has been conducted by the author unless indicated otherwise.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The subject of this research is Word Sense Disambiguation (WSD). This is the process of automatically assigning a sense to an ambiguous word in a sentence, where the choice of possible senses is determined from a lexicon.

"Word sense disambiguation involves the association of a given word in a text or discourse with a definition or meaning (sense) which is distinguishable from other meanings potentially attributable to that word."

[Ide and Veronis, 1998]

The chapter proceeds by examining the importance of WSD, this is followed by a methodological introduction which sets the context for this work. A plan of the organisation of the thesis is then given.

1.1 Importance of WSD

Natural Language Processing (NLP) is concerned with understanding a language, in our case English. WSD is an important component of this process. For example, the following sentences taken from newspaper headlines, show how the entire meaning of the sentence can change by the incorrect choice of a sense.

- Deaf mute gets new **hearing** in killing.
- Quarter of a million Chinese **live on** water.
- William Kelly was **fed** secretary.

Despite its importance, WSD is only a component, or sub-task, of a NLP system [Wilks and Stevenson, 1996]. There is little non-linguistic interest or commercial viability in a system which solely disambiguates words. Any NLP task which is sensitive to lexical ambiguity is subject to benefit from accurate WSD [Fujii, 1998]. The specific ways in which WSD can aid some of these real NLP tasks are now considered. A more detailed account is given in [Kilgarriff, 1997b].

Machine Translation

Machine Translation (MT) is one NLP task where the effects of inaccurate ambiguity resolution can easily be identified. The need for ambiguity resolution within MT has been a long standing problem [Bar-Hill, 1960]. The choice for a word in the target language will largely depend on the sense chosen in the source language. Two types of ambiguity are identified [Hutchins and Somers, 1992]. Monolingual ambiguity is concerned with the ambiguity contained in the source language. For example, the *fly in the sky* sense of *plane* translates to the French word *avion* and the *smoothing wood* sense translates to *robot.* The second type of ambiguity is called translational ambiguity. This is concerned with one sense in the source language translating to several different words in the target language. For example, one sense of the English word *ice* possesses eleven different senses in Icelandic. Therefore, WSD is solely able to aid monolingual ambiguity.

The importance of WSD to MT is highlighted by the WSD researchers who have come from a MT background [Brown *et al.*, 1991] and [Yngve, 1955]. Despite this MT systems do not apply current state of the art WSD techniques.

Information Retrieval

Information Retrieval (IR) is concerned with selecting appropriate documents from a database based on a query. IR is a well established NLP task and has become of particular importance due to the rapid growth of the internet. Lexical ambiguity is of importance in both the query and the documents to be retrieved. WSD faces a difficult challenge in both of these environments. Typical queries, particularly for the internet are very short [Grefenstette, 1997]. Therefore, there is little context available to aid the WSD process. The documents themselves must be analysed at speed in order to return results in an acceptable time frame. Therefore, efficiency requirements constrain the depth of the semantic analysis which can be performed on the document. Instead most current IR systems rely on stochastic techniques based on the lexical items, but do not consider the linguistic structure. WSD can therefore perform the role of disambiguating these lexical items. As a result, WSD may be of benefit to IR not as a component within a larger NLP system, but as an efficient alternative to performing deep semantics [Kilgarriff, 1997b]. Many WSD systems have been developed specifically to benefit IR [Voorhees, 1993] and [McRoy, 1992].

Some work has examined the benefits of WSD for IR. [Krovets and Croft, 1992] sense tagged a corpus manually so that all sense choices were correct. They discovered only a 2% improvement in the IR performance. [Sanderson, 1994] used pseudo-words to artificially introduce ambiguity into the corpus. He found that for long queries, considering all senses of ambiguous words caused no degradation to performance as there were sufficient other words to identify the required topic. However, he also found that incorrect ambiguity resolution did cause a substantial reduction to the IR performance. Finally [Schütze and Pedersen, 1995] performed only very coarse grained sense disambiguation as this is more appropriate for identifying the correct topic information. Performing this disambiguation was found to increase IR performance by 4.3%.

Text to Speech Processing

Speech processing considers the problem of generating speech from text. Some words can be pronounced in different ways depending on the chosen sense. For example, two senses of *lead*; *the type of metal* and *used for walking a dog* are pronounced differently. The correct sense choice needs to be made in order to synthesize the correct pronunciation [Stevenson, 1999].

WSD may also be beneficial for other internal components of a NLP system for example, parsing [Lytinen, 1986]. The problem of attaching a prepositional phrase relies on a semantic knowledge of the lexical items. This can only be achieved if these items are not ambiguous. However, some WSD techniques in particular selectional restrictions require a knowledge of the syntactic structure of the sentence. This therefore becomes a circular problem with interdependencies between both subtasks. A method proposed to resolve both forms of ambiguity in unison is given in [Lytinen, 1986].

As yet no quantifiable measure is available to ascertain the contribution WSD is able to make either directly for any NLP task or indirectly by aiding another component. The only way in which this can be achieved is by integrating WSD into a larger system. However, this can only be achieved if the WSD system is considered sufficiently credible to warrant integration.

1.2 Methodological Introduction

Before an analysis of the WSD problem can be considered, the context of this work needs to be established by discussing important background methodological issues. The area of this research is Natural Language Engineering (NLE) which is a rapidly growing field within Artificial Intelligence (AI).

This section discusses the general methodological issues by contrasting AI with Cognitive Science. More specific methodology adopted in this work is described by examining Natural Language Engineering. This is put into context by contrasting

it with Computational Linguistics.

1.2.1 Artificial Intelligence

There are many definitions of Artificial Intelligence (AI), this work uses the following:

> *...the field of research concerned with making machines perform tasks which are generally thought of as requiring human intelligence*
>
> [Beardon, 1989]

The most challenging fields of AI seem to be those which humans take for granted that they can perform. For example, walking, reading and writing. Tasks which humans find more complex such as long division or a database search are often much less challenging for a computer. There are two distinct reasons why developing computers to do intelligent tasks is of interest.

- To use computers as a tool to artificially simulate the human brain.
- To increase the functionality of computers.

Psychologists, linguists and philosophers want to make computer systems which will purely be used to test theories about the brain. This research interest is known as Cognitive Science. Cognitive Science restricts itself to only using methods which are employed or thought to be employed within the human brain. This approach tends to lead to the development of very small scale systems designed to test a theory rather than be of any practical use.

Artificial Intelligence aims to make computers systems intelligent for the later reason, and that is the reason behind this work. Artificial Intelligence is already being incorporated into many of the everyday applications we use, and in some cases take for granted.

- Spell and grammar checker in a word processor.
- Computer games eg chess.
- Predicting financial strategy in business.
- Performing the dangerous or unskilled jobs in manufacturing.
- Automatic diagnosis and management of treatment in health care.

Indeed the goal posts for AI have moved substantially, and at one stage the introduction of an automatic dish washer was considered intelligent.

AI systems are designed to work on a real life scale, and deal with all the external problems faced with working in a real life environment. The methods used by humans are one possible approach which may be used as a starting point or when other methods seem less appropriate. However, AI does not restrict itself to only using this method, and a wide range of techniques have been developed which have no human correlation. Therefore, the challenge of AI is, by understanding the specific problem and the resources available, to chose the best AI technique/s for that specific problem. This freedom to use which ever method seems appropriate, lifts the upper limit on performance above that which can be achieved by a human. In some areas AI systems already out perform humans, for example night vision systems.

1.2.2 Natural Language Engineering

The more general field of Natural Language Processing is the study of computer systems for understanding and generating language. By doing so it aims to develop applications which will help humans better cope with their complex environments. These applications include:

- Machine Translation
- Information Retrieval

- Information Extraction
- Dialogue systems
- Speech synthesis

The work described in this thesis has been designed and developed according to the principles of Natural Language Engineering (NLE). NLE follows on from Linguistic Engineering which is defined as follows:

> *"Linguistic Engineering (LE) is an engineering endeavour, which is to combine scientific and technological knowledge in a number of relevant domains (descriptive and computational linguistics, lexicology and terminology, formal languages, computer science, software engineering techniques, etc.). LE can be seen as a rather pragmatic approach to computerised language processing, given the current inadequacies of the theoretical computational linguistics."*
>
> [EC, 1991] page 7

The NLE approach to NLP is a pragmatic one, which specifically considers the difficulty of the task [Boguraev *et al.*, 1995]. The NLE approach sets out typical engineering criteria, found in many other disciplines, which enables it to cope with the complexity of the computer systems developed. The current level of acceptability for each of these criteria varies for different NLE tasks. The criteria considered along with the level of acceptability for this WSD task is described:

Scale The purpose of NL systems is to be able to process real-life, free text. In order to achieve this, the number of entries in the lexicon must be large scale so that all words found in the text are contained in the lexicon. Also, the system must not impose any restriction on the length of a sentence or discourse.

Robustness The system should be robust enough to handle free text in any domain. The system should not crash or be badly affected when encountering difficult circumstances such as very long sentences, or words not found in the lexicon.

Maintainability The system should be useful over a long period of time. In order to achieve this, it must be flexible to change. Also, as the personnel developing a system are likely to change over a long period of time, all code written must be developed to facilitate the process of other people further developing the code.

Flexibility The system should be flexible so that it is able to be adapted to operate in different domains. This refers to the topic domain of the text and also the lexicon which is used.

Integration The system should allow ease of integration with other sources of knowledge and facilitate the process of being integrated into a larger system.

Feasibility The hardware requirements of the system should not be too substantial. Therefore, the system must operate at an acceptable speed during training (if required) and testing. The system should also be able to operate with an acceptable amount of memory.

Usability The ultimate criterion for success for a system is that end users are happy with it. Within the research environment, the core functionality is the most important feature of this criterion. Other features which are important in the business environment such as user friendliness and a good marketing strategy are not considered goals of NLE research.

The pragmatic approach adopted by NLE is in contrast to Computational Linguistics (CL). CL is a more theoretical study of the human language. A common criticism of applications which adopt a CL approach is the inability to process realistic material:

> "Computational linguistics research in practice tends to revolve round little "toy" subsets of artificially simple linguistic forms, in the hope that systems which succeed in dealing with these may eventually be expanded and linked together until they cover entire languages."
>
> [Sampson, 1987] page 17

NLE differs from CL by incorporating a full range of AI techniques. NLE may use CL theories when applicable, but will also make the most of what ever else is available.

1.2.3 Symbolic and Sub-Symbolic Approaches

The traditional approach to artificial intelligence is symbolic, involving a representation of the problem, and a mechanism to search through it. Traditionally this mechanism was considered sufficient to generate artificial intelligent behaviour.

> *"A physical symbol system has the necessary and sufficient means for general intelligent action. By "necessary" we mean that any system that exhibits general intelligence will prove upon analysis to be a physical symbol system. By "sufficient" we mean that any physical symbol system of sufficient size can be organized further to exhibit general intelligence."*
>
> [Newell and Simon, 1976]

A characteristic of many symbolic approaches is the development of rules which enable a chaining process towards an intelligent solution. This chaining process enables a chosen solution to be identified which conforms to these rules. The solution can be shown to conform to the rules as part of its reasoning which increases the credibility and level of acceptance of the choice made.

A significant challenge to the symbolic approaches has come from adaptive learning mechanisms. The initial most significant step was through parallel distributed processing [Rumelhart *et al.*, 1986]. These sub-symbolic approaches have

developed into two significant branches, stochastic approaches based on Bayesian probabilities and machine learning approaches in particular evolutionary algorithms and neural networks. The sub-symbolic approaches are based on simple components and the interaction between them. Unlike symbolic approaches, they are generally unable to provide reasoning for their solutions. However, they are characterized by an ability to learn and adapt to different environments.

> *"...it is widely believed that there are some activities of intelligence (e.g. recognition of multidimensional patterns) where an approach operating at some lower level than a level of description in symbols is more appropriate than the traditional logical-symbolic approach."*
>
> [Calmet and Campbell, 1993]

The methodology adopted in this work recognises that beneficial characteristics exist for both symbolic and sub-symbolic approaches. To restrict oneself solely to considering one approach may identify the upper limit for the technique, but may not identify the upper limit for the solution. Therefore, the work presented in the thesis adopts an engineering methodology and considers all possible approaches. The aim being to enable these approaches to complement and not contradict each other.

The benefits of combining approaches can be highlighted by considering an example. To learn to play cricket a number of rules must be learnt; the laws of the game, the fielding positions and the basic technique for batting, bowling and fielding. These could all be accomplished by a symbolic approach. Learning the game also requires extensive practice to experience many times over the different situations which may arise. This is equivalent to a sub-symbolic approach. Therefore, to become good at the game both symbolic and sub-symbolic techniques are required.

1.3 Logical Progression of the Thesis

This section sets out the framework by which the thesis is organised. This is carried out by summarising the issues considered in each chapter.

Chapter 1 sets out the problem definition for this work and discusses the importance of WSD. Methodological issues are then addressed in relation to the position of this work within the field of computer science. Finally a plan of the thesis is given.

Chapter 2 considers the main challenges within WSD so that a better understanding of the problem can be established. In particular, the chapter concentrates on the various resources available to aid WSD and the problems associated with evaluation. The criteria for success is then given.

Chapter 3 considers other approaches to the task of WSD. The chapter is structured by considering various knowledge sources adopted to aid WSD. The features of these knowledge sources are reflected by the characteristics of the systems which use them. The problems associated with evaluation within WSD prevent a detailed comparison of the systems considered.

Chapter 4 commences the discussion of DURHAM, the system developed aiming to fulfil the criteria for success. The chapter examines the knowledge sources adopted by DURHAM to provide information to aid disambiguation. The chapter introduces a novel knowledge source named contextual information and examines the way this is learnt. The chapter also considers the method by which the knowledge sources are combined.

Chapter 5 examines the mechanism used to calculate the scores for each knowledge source. The chapter then progresses to consider the disambiguation algorithm developed to select a sense for each ambiguous word. The disambiguation algorithm is novel, and provides a compromise between accuracy and efficiency.

Chapter 6 initially sets out various evaluation metrics which are used to assess the performance of DURHAM. These evaluation metrics are then used to report the results achieved by DURHAM for the large scale task of disambiguation on SEMCOR. Analysis of DURHAM is then performed to discover the effect various components of the system have made to results. The work is then compared with another system which has been evaluated on the same test set. A more complete comparison with other systems is not possible due to other work being evaluated on different data sets. However, an analysis of the feasibility of comparing WSD systems performing on different data sets is given.

Chapter 7 reports the evaluation of DURHAM on the SENSEVAL task. The chapter examines the differences between evaluation on SEMCOR and SENSEVAL and the various adaptations to the system required. In particular a further knowledge source is introduced into DURHAM named clue words. The results achieved are reported and compared with other systems which also took part in the evaluation. A discussion is then given concerning the scalability of clue words and the SENSEVAL evaluation.

Chapter 8 provides a conclusion to the thesis by considering the criteria for success. The ability of the system to meet these criteria is discussed. Various directions in which work in the future could build upon this system are also discussed.

Appendix A A list of the SEMCOR files used for training, testing and blind testing.

A Glossary of terms is also provided.

Chapter 2

The Word Sense Disambiguation Problem

2.1 Introduction to WSD

Now that the WSD task has been defined at the start of chapter 1, this chapter will examine the challenges faced by the developers of WSD systems. The challenges considered are generic to all WSD systems and do not consider any further specific problems encountered in this work. The chapter highlights the problems with context information, using a lexicon as a sense inventory and as a knowledge source. The chapter also considers the difficulties of training and testing WSD systems. Only once the problems within the field have been clearly examined will a detailed criteria for the success of this work be given.

WSD is no more difficult than part-of-speech tagging [Wilks and Stevenson, 1996]. This claim made by Wilks and Stevenson suggests that 92% accuracy can be achieved for disambiguating all open class words[1]. They claim that this high accuracy can be achieved solely by knowing the frequency distribution of the senses and correctly identifying the part-of-speech. However, it is easy to misinterpret

[1] Open class words are nouns, verbs, adjectives and adverbs. Closed class words are determiners such as *the, of, in, a etc* and are not generally considered as ambiguous

the high accuracy achieved. The metric adopted to test the system includes words which are not ambiguous and uses a coarse grained lexicon.

Using the WordNet lexicon, only 62.1% accuracy will be achieved for ambiguous words if the most frequent sense belonging to the correct POS is always chosen. This result shows that WSD is a more difficult task than POS tagging. The reason for this is that WSD must attempt to categorize a word to a much finer level than is required for POS tagging. Moreover, the WSD categories (senses) are different for each word, whereas they remain the same for POS tagging.

2.2 Context Information

Even if it is accepted that WSD is more difficult than part-of-speech tagging, the difficulty of the task is generally still not fully appreciated. *"Word sense disambiguation is easy - you just look at the context!!"* - to quote the typical first impressions of someone considering the problem! It is true that the context of an ambiguous word is important for resolving the ambiguity. However, the following examples taken from [Hearst, 1991] show how the sense of the word *tank* changes despite many words in the sentence remaining the same.

- "Plagued by a critical shortage of fuel tanks".
- "Plagued by a critical shortage of fuel and tanks".
- "Plagued by a critical shortage of fuel and tanks, the army were unable to advance".

In the first sentence, the sense of *tank* is clear as *fuel* takes the role of a modifying noun describing the type of tank being referred to. However, in the second and third sentence *fuel* is a noun and in this role provides less conclusive evidence for the *fuel tank* sense. Hearst identifies this difference and aims to take advantage of local syntactic information to benefit more fully from the surrounding context. However, the *fuel tank* sense is still the most likely meaning of *tank* in sentence

two. In sentence three the context word *army* provides strong evidence for the *military* sense of *tank*, although there is insufficient context to ensure it is referring to this sense. Despite this, it has been shown in sentence two that *fuel* provides evidence for a different sense. Therefore, in order to achieve the correct ambiguity resolution, the system must be able to weigh the value different parts of the context are able to provide for different senses. In this example, *army* should be identified as stronger contextual evidence than *fuel.*

The next set of examples are all taken from newspaper headlines. They show that the syntactic information is not always sufficient, and only deep semantic knowledge will resolve the ambiguity. The examples demonstrate how the incorrect resolution of an ambiguous word (given in bold) can significantly alter the meaning of a sentence. The first two examples also demonstrate the circular problem caused by using the information from the context of the sentence. To resolve the ambiguity of a word requires the knowledge of the meaning of the context, but this too can be ambiguous. Therefore, there is a problem of knowing which ambiguous word should be resolved first.

- Iraqi **head** seeks **arms**.
- Farmer **bill** dies in **house**.
- Police begin campaign to **run down** jay walkers.
- Milk drinkers are **turning** to powder.
- Two convicts evade noose, jury **hung**.

2.3 Lexical Problems

Although a number of difficulties of using contextual information have been shown, understanding the context in which a word is used is essential to enable accurate ambiguity resolution. Developing ways to best exploit this information is the challenge faced by WSD researchers. However, the lack of resources available to help

disambiguation is a major problem hindering this process. The most important resource which is available for researchers is the lexicon.

The lexicon is able to provide two important roles within WSD. The first role is as a sense inventory to provide the list of senses which the WSD system must distinguish between. The second role is as a knowledge source to help understand the context in which a word is used to aid ambiguity resolution. Difficulties with the lexicon performing both of these roles are now considered. It is important to consider these difficulties so that the correct choice of a lexicon can be made by the WSD developer. However, it is not the aim of this work to try to find any solutions to these difficulties. The challenge of developing a lexicon is the research interest of lexicographers [Hanks, 1993] [Kilgarriff, 1993].

2.3.1 The Lexicon as a Sense Inventory

Problems exist for WSD developers regardless of which lexicon they choose as their sense inventory. The problem stems from the fact that no lexicon has been designed specifically for the WSD community. Therefore, the procedure of how to split a word into its component senses is dependent on the objectives of the lexicon. This causes a large variation in the sense divisions between different lexicons. The difficulties involved with the task of assigning a sense to a word are now considered.

It is a characteristic of the English language that most frequently used words are ambiguous. As these words are well understood, they are able to be applied in several different contexts without causing confusion. It is their usage in these different contexts which generally leads lexicographers to distinguish between them and define individual senses for each context. For example the adjective *brilliant* is usually used to describe something which is pleasant. It is frequently used within different contexts to describe a performance, smile, musical note or a light. In most lexicons *brilliant* is assigned a separate sense for each of these contexts. Not only does this greatly increase the number of senses, it also creates many senses which are extremely similar in meaning.

The Bank Model

The difficulty of assigning senses to words is a greatly discussed topic by lexicographers. The discussions often adopt *bank* as an example from which the **Bank Model** is derived. The Bank Model primarily is designed to show that some senses of a word are homonymous with no relation between them. However, this model can be shown not to generalise for all words [Kilgarriff, 1992]. The *bank* example can be adopted to highlight the difficulties facing lexicographers in defining senses for a word.

Two noun senses of bank come immediately to mind, one concerning money and the other bordering a river. It is argued in [Hanks, 2000] that these two senses are actually two different words which are spelt the same way. Analysis of the derivatives of each sense adds weight to this argument. The *side of the river* sense is derived from medieval French (banki) or old Icelandic (bakki). The *money* sense comes from medieval Latin (bancus/a).

Sentences can be created such as *"I ran to the bank"* which could be referring to either sense/word. However, these types of sentences rarely occur in real text. Some context which determines the sense being referred to usually exists in real text. For example, bank balance, bank manager, rob the bank, or slippery bank, burst its banks and flood banks.

However, artificially created sentences for which the ambiguity remains unresolved is not the difficulty highlighted by the Bank Model. For many words, including bank, one sense shades into another, capturing some but not all of the features of the initial sense. For example *blood bank* and *sperm bank* share some features which are similar to the financial sense of *bank*. They are all institutions responsible for the safe keeping of an object. Equally *sand bank* and *bank of snow* have features similar to the *river bank* sense, but neither are by the river. The next examples show that the problem of assigning senses to words can become even more difficult. Seemingly the same sense of bank is referring to three very different things.

- "The bank will be closed on Saturday."
- "The bank has made a mistake with my overdraft."
- "The bank needs a large refurbishment."

The first example refers to the bank as an establishment, the second to the people working in the establishment and the third to the building in which the bank is housed. Bank is by no means an exception. Many other words have similar features, and *television* may be considered as a further example:

- "Put the picture on top of the television."
- "The television has inspired me to do great things."
- "Television has injected large amounts of money into the game."

It is the role of lexicographers to decide whether to create a new sense for a word or to accept that a sense can be used for a slightly different meaning. This process is known as *lumping or splitting.* Some lexicographers prefer to lump senses together, generalising the details and invariably making the definition more vague. Other lexicographers try to be precise, splitting words into many senses in order that each sense can only be used in a single context. Ultimately, the policy adopted depends on the requirements of the dictionary being produced. As a result, there are large discrepancies between sense distinctions in different lexicons.

A proposal for classifying the similarity between senses is given in [Miller and Teibel, 1991]. Miller and Teibel propose three categories of similarity:

- **Categorical** - for senses which have a different part-of-speech.
- **Homonymous** - for senses which have completely different meanings.
- **Polysemous** - for senses which are similar.

However, this model does not give a true representation of the problem. For many words there exists graded levels of similarity between senses. As a result, no clear cut off point between polysemous and homonymous senses exists.

Task Specific Lexicons

One of the reasons identified why there are difficulties in choosing the correct lexicon for WSD is that no lexicon has been designed specifically for WSD. However, difficulties would remain even if a group of lexicographers were assigned the task of preparing a standard lexicon specifically for WSD. In reality, the WSD community would not be able to agree the degree of granularity to which the lexicon should be defined. WSD is an internal task within NLP, and different real tasks require different lexicons. Therefore, the appropriate WSD lexicon is dependent on the NLP task it is aiming to assist. Whilst the distinction between two senses of a word may not be important for one NLP task, it may be very important for another. For example, consider three senses of *band* - a musical band, a radio frequency and a rubber band. The first two senses both translate to the same Italian word *banda*. The rubber band sense translates to a different Italian word *cerotto*. Therefore, a NLP system attempting to translate from English to Italian would not need to use a lexicon which makes a distinction between the first two senses of *band*.

One possible solution may be to develop a dictionary which splits senses to such a fine level that it is sufficient for all NLP tasks. If a specific task only requires a vague meaning of a sense, then choosing the wrong sense from the lexicon may not degrade the performance of the larger task. Using the above example if the *radio band* sense was chosen instead of the *music band* sense, the correct translation would still be made. There are two problems with this approach. Firstly the machine readable dictionary would become very large. It is unlikely that any single NLP task would require anywhere near the level of detail contained in it. The second problem is more important particularly from a WSD perspective. To improve a WSD system and to integrate it into a NLP system, it is important to be able to test the accuracy of the WSD system. To do this a mechanism to evaluate the disambiguation algorithm is required. Testing a disambiguation system is performed using a corpus of manually sense tagged text. If the lexicon is too finely grained, then even humans will find it difficult to accurately assign the correct sense to a word. Therefore, manually sense tagged corpora become

very difficult to produce. If the agreement of sense choices between different sense taggers is low, then the credibility of the sense tagged corpus is reduced. The problems of manual sense tagging are discussed in section 2.4.

Effect on WSD

This section has identified three difficulties with the way words are split into their component senses. Their effect on WSD is now considered. The first difficulty is the semantic similarity between different senses of the same word. This makes the WSD process much more difficult because semantically similar words are more likely to be used in similar contexts. Therefore, a very accurate understanding of the context is required before the correct choice of two similar senses can be made.

The second difficulty is the large number of senses which are assigned to many words, in particular the frequently occuring words. The large number of possible senses leads to an explosion of the total number of possible sense combinations for a sentence. For example, even in a short sentence *"The boy will be on a run before school starts"*, there are 5,754,112 different sense combinations for that particular sentence, using the WordNet lexicon. Even if the correct POS is known for each word there still remains 348,480 sense combinations. This explosion of sense combinations is known as Wilks' problem [Slator and Wilks, 1987]. Wilks' problem shows the difficulty of correctly disambiguating an entire sentence. Moreover, if the system is going to perform at an acceptable speed, it highlights problems associated with the computation time which can be spent on each possible sense combination.

The third difficulty identifies the large variation between the way different lexicons have chosen to assign senses. This prevents the WSD developer from employing more than one lexicon. Multiple lexicons could however be beneficial to enable one lexicon to be used as the sense inventory and further lexicons to be employed as a knowledge source. Moreover, the variation between lexicons prevents an evaluation between different systems which use different lexicons as their sense inventory. The problems of evaluation are considered further in section 2.5. The

variation between lexicons also causes difficulties for using a lexicon as a knowledge source. This role of the lexicon is now considered.

2.3.2 The Lexicon as a Knowledge Source

In addition to the lexicon providing a list of senses for the WSD system to distinguish between, the lexicon is also able to serve as a knowledge source to aid disambiguation. Most lexicons provide a definition with each sense that is listed. This definition enables the reader to distinguish between all the possible senses of that word. A great deal of research has concentrated on automatically extracting information from these definitions which can aid disambiguation. This work is discussed in section 3.2. However, this section examines some of the problems with the dictionary definitions resource.

Until recently, lexicons were not available in machine readable format. Lexicographers developing dictionaries to be published needed to consider the size of the dictionary being produced. As a result, the length of the definitions were constrained to the minimum so that the reader was still able to distinguish between the possible senses. This compromises the precision and uniformity with which the senses are defined. As a result, the automatic extraction of information from these definitions is very difficult. More recently machine readable dictionaries have become available which removes the size constraint. However, the problem of clearly being able to make distinctions between senses still exist.

Some dictionaries, including the *Cambridge International Dictionary of English* (CIDE) [Procter, 1995] and the *Longmans Dictionary of Contemporary English* (LDOCE) [Procter, 1978], have adapted their dictionaries, which has made them more useful for NLP. All definitions in the dictionary are made up from a core of 2000 words, predominantly to aid foreign users of the dictionary. However, the process of automatically extracting semantic information from the dictionary definition is substantially facilitated if only a core 2000 words are used in the definition. As a result, it is possible for the computer to acquire knowledge about other words outside the core from their definitions [Poria, 1999].

A further problem still exists with using the dictionary definitions as a knowledge source. The definitions are designed to enable a distinction between the possible senses in that lexicon. However, as already discussed the variation in the way words are split into component senses is very different for each individual lexicon. The type of information required to make sense distinctions for one lexicon may be very different for a separate lexicon. Therefore, the ability for this knowledge source to be applied to a different lexicon is reduced.

2.4 Sense Tagging and Inter Tagger Agreement

A corpus of manually sense tagged text is an essential resource for WSD systems. The corpus provides a mechanism for testing, enabling different algorithms to be compared. Moreover, some disambiguation algorithms require training data in the form of manually sense tagged text. For these systems, the choice of lexicon may be restricted to those for which sense tagged data is available. This section examines some of the problems concerned with developing a manually sense tagged corpus.

The quality of a manually sense tagged corpus can be measured by its accuracy. The metric for computing accuracy is the Inter-Tagger Agreement (ITA), which is applicable so long as more than one person has sense tagged the same text. The ITA is defined as the percentage of words which have been assigned the same sense by all sense taggers. The ITA is perceived as an upperbound figure for WSD systems [Kilgarriff, 1998a]. A computer system would not be expected to achieve higher accuracy than a human, as is possible in some other fields within AI. Also if a disagreement amongst the sense taggers exists there may be errors present in the corpus which the computer system is being tested against.

It would be easy to conceive that sense tagging for a human is a simple task, and as a result, all humans always agree on the sense of a word. Unfortunately this is not the case. The remainder of this section discusses the main causes for disagreement amongst taggers together with ways in which this disagreement can be minimized.

2.4.1 Dictionary definitions

The people performing the sense tagging are often not the same people who developed the dictionary which is being used as the sense inventory. This is a major cause of disagreement amongst senses taggers. The way in which a lexicographer splits a word into its component senses may seem unusual to the sense tagger who perceives the senses of a word differently. The dictionary definitions are normally designed to give general indication of the meaning of a sense. In instances where one sense shades into another, the definition is unlikely to be be able to provide a definitive indication of where one sense stops and another begins. As a result, the tagger is left with doubts as to the exact meaning of each sense. When there is doubt then each tagger will use their intuition to determine what each sense refers to. Individual taggers may resolve this doubt in different ways, leading to disagreement. Resolving any uncertainties before tagging commences would be the simple solution to the disagreement. Equally important though, is that the taggers should primarily work in isolation from each other. Being isolated ensures that each set of results has not been influenced by another tagger. This bias would undermine the value of a corpus which has been multiply sense tagged. Many recently published dictionaries have taken steps to approach the problem of clarity to the fine level required. The ability to be able to distinguish between different senses has been improved by accompanying the definitions of senses with examples of their use.

2.4.2 Dictionary feedback

In previous years, writing dictionaries and using them to sense tag data were considered two separate tasks. The taggers would have to accept and work with the sense distinctions they were given. No mechanism existed allowing taggers to suggest changes to definitions or sense distinctions. CIDE is one of the current dictionaries which has enabled an iterative process. Problems encountered whilst sense tagging are referred back to the lexicographers responsible for writing the dictionary. By returning to these lexicographers, sense distinctions may be revised and dictionary definitions may be made clearer. All amendments should make the

sense taggers task less ambiguous, leading to higher ITA.

With an iterative approach the sense tagged corpus can be viewed as a test bed for the development of the dictionary. The necessity for manual sense tagging for dictionary development is important. WSD is not a large enough field by itself to warrant the investment of funds required to produce a sense tagged corpus. This helps to ensure that more sense tagged corpora will become available which is of great benefit to WSD systems.

An example of how sense tagging is able to offer valuable information for lexicographers is given in [Bruce and Wiebe, 1998]. Five people were assigned the task of sense tagging 2369 instances of the word *interest*. The LDOCE dictionary was used as the sense inventory, from which 6 senses of *interest* were considered. In general the ITA between the five judges was very high. However, analysis of the results showed that most of the tagger's error occurred between two of the senses. The two senses were closely related, and the taggers found it difficult to distinguish between them. This information could be used by a lexicographer, who may then decide to lump the two closely related senses together and give a broader definition to encompass both senses. Combining the two senses increased the Kappa[2] value for the ITA from 89.8% to 91.6%. In this example, Kappa is a useful metric for measuring ITA. After the two senses have been combined the number of agreements expected by chance increases. Kappa is able to take this into account enabling a fair comparison of ITA before and after the two senses are combined.

2.4.3 Sentence Ambiguity

Even if no uncertainty exists in the definition of the sense, some sentences will continue to cause problems for the taggers. Some of the examples offered to highlight these difficulties are taken from [Krishnamurthy and Nicholls, 2000]. Krishnamurthy and Nicholls' task was to produce a manually sense tagged gold standard for the SENSEVAL evaluation. The SENSEVAL evaluation is discussed in section

[2]The achieved accuracy relative to the accuracy which can be achieved by chance. Kappa is detailed in section 2.5.2

2.6.

Many senses are defined by a collocation or a multi-word expression. Taggers need to decide the degree of variation allowed in these expressions. For example, it may seem reasonable to assign the expression *couldn't be bothered* to the *can't be bothered* sense which is the defined phrase. But what about *can be bothered*? Should *knees of jelly* be assigned to the *weak at the knees* expression? These decisions are arbitrary and difficult to lay out rules for all cases. As a result, the decision must be left to each individual tagger.

Another problem occurs when the context of a word is insufficient to conclusively determine the sense of the word to which it is referred. Some sense tagged corpora try to accommodate this by allowing the tagger to list all possible senses. For example, in producing the gold standard for SENSEVAL, distinctions needed to be made between two closely related senses of *bet*. One sense refers to the gambling sense where money is wagered - *"I put a bet on the third race at Aintree"*. The other sense is a verbal speculation - *"I bet he doesn't get here on time"*. In these two examples the distinction is clear, but in other cases it is less so.

- "I bet Owen will be the first to score."
- "I lost my bet today."

In both of these examples, it is not clear whether money is wagered. Therefore, the tagger should list both the possible senses. However, using world knowledge a tagger may decide that the sense being implied can be inferred without listing both possible senses. This subjective decision provides another source of inconsistent sense tagging. A similar problem is reported in [Wiebe *et al.*, 1997] where the task is to tag common verbs with their WordNet senses. In their example - *"The group has forecast 1989 revenue of 56.9 billion francs."*, the sense for the verb *have* can not be determined. Has the group done the forecasting? or has a third party forecasted the group's revenue?

2.4.4 Skilled sense taggers

As sense tagging is a very labourious process, a temptation exists to employ unskilled people for the task. However, experience has shown that the quality of the sense tagged corpus is degraded if inexperienced lexicographers are employed.

The DSO corpus is sense tagged with WordNet meanings by a group of undergraduates in Singapore [Hwee Tou Ng and Hian Beng Lee, 1996]. They adopted a lexical approach, by sense tagging all instances of the most frequently occurring nouns and verbs in the SEMCOR corpus. 121 nouns and 70 verbs were considered, and 1,500 instances were sense tagged for each word. Frequently occurring words typically have the greatest number of senses, making the task more difficult. Also English was not the first language for many of the sense taggers. As a result, understanding the differentiation between senses proved very difficult. Both the DSO and SEMCOR corpuses contain text taken from the Wall Street Journal which enables comparisons to be made. On this subsection of the corpus the agreement between the DSO sense taggers and the SEMCOR sense taggers (ITA) was reported to be 57%.

The SEMCOR group themselves have investigated the importance of using skilled lexicographers to perform sense tagging [Fellbaum *et al.*, 1996]. They found that on average the 'naive' taggers agreed with the experienced taggers in 74% of all instances.

Bruce and Wiebe's work described in section 2.4.2, also highlights the effect of using skilled lexicographers. Out of the five judges who sense tagged the word *interest*, only two of them were experienced sense taggers. The two experienced taggers agreed with each other in 96.8% of all cases. The unskilled taggers showed biases towards different senses resulting in lower ITA figures. The lowest ITA between two of the unskilled taggers was 88.4%.

2.4.5 Textual or Lexical

Two approaches exist for manually sense tagging a corpus.

- A textual approach assigns a tag to every context word in a sentence.
- A lexical approach chooses a set of words to be sense tagged. For each of these words, sentences are selected in which the word is contained. Only that particular word is sense tagged in the sentence.

Both of these approaches are discussed.

SEMCOR is the most widely used and best known manually sense tagged corpus [Landes *et al.*, 1996]. It is sense tagged using the WordNet lexicon [Fellbaum, 1997]. Most of the text is taken from the Brown Corpus, which is made up of extracts from the Wall Street Journal. The remaining text consists of Stephen Crane's novel *The Red Badge of Courage* [Crane, 1987]. SEMCOR is a textual corpus, therefore every open class word in every sentence is sense tagged. SEMCOR can be contrasted with the lexical SENSEVAL corpus. The SENSEVAL corpus sense tags 35 different words, these words include nouns, verbs and adjectives.

Potentially a textual corpus is a comparably more valuable resource than a lexical corpus. A textual corpus encourages a large scale approach to word sense disambiguation, and is more applicable to real NLP tasks. Moreover, a textual corpus enables the evaluation of the effect of using the correct sense of the context words.

However, it is difficult to obtain a high ITA with a textual corpus, and this its main disadvantage. The sense taggers compiling a textual corpus are faced with a more difficult task than those compiling a lexical corpus. Textual sense taggers are unable to concentrate on one particular word at a time. The taggers must constantly read and understand the sense distinctions of each word in the text as it is sense tagged. When a word reoccurs in the text, the sense taggers interpretation of the sense distinctions may vary, leading to inconsistencies. The lexically tagged SENSEVAL corpus achieved a 96.5% ITA, much higher than any of the ITA figures mentioned earlier on SEMCOR.

The process of continuous reference to the lexicon makes the production of a textual corpus extremely labour intensive. In relative terms, a lexical corpus is

less labour intensive. [Krishnamurthy and Nicholls, 2000] report that on average the SENSEVAL sense taggers achieved a speed of 66 instances of a word per hour. With an increase in the number of instances they sense tagged, the taggers became increasingly familiar with the sense distinctions. This enabled the tagging speed to increase. In addition, the taggers found that it was often only necessary to look at the immediate context of the word. This also increased the tagging speed.

The final problem with a textual approach is the low number of instances of a particular word. In SEMCOR a great number of words are sense tagged, but less than 100 senses occur more than 500 times. A WSD system which requires training data is likely to prefer a lexically sense tagged corpus, in order that many examples exist of each word being considered.

2.4.6 Automatic Sense Tagging

This section has detailed many of the problems associated with manually sense tagging a corpus. Therefore, is it possible to generate a corpus of sense tagged data automatically? This question is considered in [Gale and Church, 1991a] and [Gale and Church, 1991b]. Their method is to identify parallel text written in more than one language such as the Canadian Hansards[3]. Using the word *duty*, they are able to distinguish between the *tax* and *obligation* senses of the word. This is done by examining the parallel French text to see which word is used for the translation.

For the word *duty* this method was successful. It enabled a statistical based disambiguation system to be developed which used the automatically tagged data for training and testing. The system is detailed in [Gale *et al.*, 1995]. However, for most words such a method would not be possible. The method relies on the word having completely distinctive senses so that each sense maps to different words in the target language. The system reported in [Brown *et al.*, 1988] and [Brown *et al.*, 1991] also uses the Canadian Hansard as training data. However, inability to make fine grained sense distinctions does not cause a problem for this system. The reason

[3]The Canadian Hansards are proceedings from the Canadian Parliament which are published in both English and French.

for this is that Brown's system was developed specifically for MT and therefore the Canadian Hansard is able to provide the correct level of sense granularity for this task. Other problems, which both systems must consider, are concerned with identifying the parallel word in the text. Since in any language there are many ways to say the same thing, it is possible to convey the same meaning without using a particular word. Determining the correct sense would then involve a deep understanding of the target text.

2.5 Evaluation Difficulties in WSD

The desire to compete and test oneselves against others is a natural human instinct. This instinct can be seen in most aspects of our daily life. At work we are assessed, and pay may be dependent on performance. Many people partake in some kind of sport where one trains to compete against others. This competitive natural instinct is a very healthy one. The desire to do well provides us with the motivation to drive ourselves in order to succeed.

The importance of evaluation has been shown in many areas of NLP. Performance of state-of-the-art POS taggers and parsers are often acceptable for the task required, although neither are completely error free. Both of these tasks benefit from having common resources for training and testing, most significantly the Penn Treebank corpus and the Brown corpus. In addition to a common corpus, these fields also benefit from standard evaluation metrics. These two factors enable an evaluation mechanism to be established within which different systems can evaluate on a common task, leading to the generation of a worthy comparison between systems.

Government funding agencies have also recognised the importance of evaluation. The MUC competition [Kaufmann, 1995] has enabled other NLP tasks such as information extraction, proper-noun classification and even anaphora resolution to possess a framework for evaluation. In addition to providing a motivation, the MUC evaluation also provides inspiration. MUC facilitates the opportunity to pool

ideas, and discuss common problems by providing a focus towards a common task. An extensive study of evaluation within NLP is given in [Callaghan, 1998].

2.5.1 WSD Evaluation

Unfortunately, there are many problems involved with setting up an evaluation mechanism for WSD systems. This section discusses some of these difficulties.

One of the fundamental principles behind all worthwhile competitions is that it must be fair for all competitors. This may seem intuitive, but for WSD this is extremely difficult to achieve. Different systems are developed using different sense inventories. Each sense inventory will split senses and give definitions in a unique way. The word *interest* is used to highlight the difference. LDOCE has ten senses of *interest*, WordNet has eight, and [Zernik and Jacobs, 1990] consider only four senses of the word.

It is important that the evaluation mechanism provides free text. This free text must be sense tagged with one particular sense inventory. Any system which has been developed using the same sense inventory as used in the evaluation will immediately hold an advantage. This is because the system will have been developed to take advantage of the information which that particular lexicon offers and tailored to the specific sense inventory. Information given in other dictionaries used by other systems may not be available.

By providing mappings between the senses of different lexicons, attempts to overcome the use of a wide variety of lexicons could be made. The mapping process is equivalent to sense tagging the text with each of the different sense inventories. However, if all systems were allowed to assign a sense tag from the lexicon familiar to them those systems using a coarsely grained sense inventory would benefit more than those systems which did not. Secondly, developing a mapping between lexicons causes information to be lost. Senses overlap, and as a result creating a complete map may not be possible. To avoid the problems, the prefered solution is to use a lexicon which is not used by any WSD system.

Even if an appropriate lexicon can be found, there are still many other problems. A successful evaluation is reliant on an accurate manually sense tagged corpora. Section 2.4 discussed the difficulties involved with producing such a corpus. The evaluation must determine a method to deal with sentences where human sense taggers disagree. Choosing to remove all these sentences would make the task artificially easier and would not give a true reflection of the accuracy achievable by state-of-the-art WSD systems. If human taggers have been able to assign multiple senses to a word, should the system be expected to produce all of these senses?

The difficulty of evaluating a WSD system is discussed in [McRoy, 1992]. McRoy made an unsuccessful attempt to quantify the performance of her WSD system. She reports the reasons for the evaluation failure being due to the difficulties of manual sense tagging. The manual sense tagging is described as *"... far more tedious than manual part of speech tagging or bracketing"* (McRoy, 1992, p. 26) In addition McRoy questioned the benefit of evaluating WSD systems as WSD is a sub task of a NLP system.

2.5.2 Evaluation Metrics

Unlike parsing or POS tagging there are no standard evaluation metrics for WSD. Different metrics used for WSD are now considered.

The Stochastic Scoring Metric

A proposal for a stochastic scoring mechanism is given in [Melamed and Resnik, 2000]. The metric enables the evaluation of stochastic systems which assign probabilities to several senses rather than choosing one particular sense. The stochastic system scores the value of the probability assigned to the correct sense of the word. The metric is also able to evaluate deterministic systems which uniquely choose only one sense. These systems score 1 if the correct sense is chosen and 0 otherwise. The following example shows that this evaluation metric favours a deterministic system.

Table 2.1: Example of score discrepancy between stochastic and deterministic system using Resnik's scoring system

Test Number	Correct Sense	Stochastic Score	Deterministic Score
1	A	0.7	1
2	A	0.7	1
3	A	0.7	1
4	A	0.7	1
5	A	0.7	1
6	A	0.7	1
7	A	0.7	1
8	B	0.3	0
9	B	0.3	0
10	B	0.3	0
Total		**5.8**	**7**

For this example, let us take a simple word which has two senses A and B. Using training data provided we find that A is referred to in 70% of the instances and B 30%. Assuming that this is the only information available, the stochastic system will assign A a score of 0.7 and B a score of 0.3 in all test cases. The deterministic system will always choose sense A as it is the more frequently occurring sense. If the frequency distribution of senses in the test data roughly reflects the frequency distribution in the training data, the deterministic system will achieve a higher score than the stochastic system, using Melamed and Resnik's scoring scheme. This is shown in table 2.1.

The stochastic system will out perform the deterministic System, only if sense B was referred to in more than 50% of the testing instances. In this instance the baseline accuracy established by choosing a sense at random would outperform both systems.

This example is not intended to suggest that deterministic systems outperform stochastic systems. The example simply highlights the difficulty of finding a scoring scheme which is able to evaluate both types of system fairly.

Kappa

Other evaluation metrics take into account different aspects of the problem. Kappa and entropy both quantify the difficulty of the task in order that the comparison of different tasks can be improved.

Kappa reports disambiguation accuracy relative to a chance baseline. Kappa is calculated as:

$$Kappa = \frac{a - e}{1 - e} \tag{2.1}$$

where a is the achieved accuracy and e is the accuracy which could be achieved by chance. A Kappa value of zero means that the system has achieved the same accuracy as the chance system, and a perfect system will score a Kappa value of one.

How to calculate the accuracy of a chance system is one of the questions which developers need to consider when using Kappa. There are two choices: Consider a chance system to be one which chooses each sense with equal probability. Alternatively consider a chance system to be one which chooses the most frequently occurring sense. The former takes into account the number of senses for each word, but neglects the frequency distribution of those senses. The later considers the most frequent sense, but neglects the number and frequency distribution of the remaining senses.

Entropy

An entropy measure is used widely within statistics to measure the confusion within a system [Charniak, 1994]. It may therefore be possible to use entropy to measure the confusion within a disambiguation system and quantify the difficulty of the task. Unlike Kappa, entropy takes into account both the number of senses and the frequency distribution of those senses. Entropy is calculated as:

Table 2.2: Table showing how entropy scores change for different frequency distributions of senses

Word Number	Frequency Distribution	Frequency Baseline	Entropy
Word 1	0.5 0.5	0.5	1
Word 2	0.8 0.2	0.8	0.7219
Word 3	0.8 0.05 0.05 0.05 0.05	0.8	1.1219

$$Entropy = -\sum_{i=1}^{n} p_i log_2 p_i \tag{2.2}$$

where p_i is the probability of sense i appearing in the text and n is the number of senses of the word. A high entropy measure reflects a word which is difficult to disambiguate. Table 2.2 gives examples of different words with their frequency distribution and entropy measure. The table shows that to some extent entropy does indeed reflect the difficulty of disambiguation for each word. Word 2 has a lower entropy than word 1, which shows that entropy has captured the effects of an uneven frequency distribution for word 2. Moreover word 2 has a lower entropy than word 3 showing that the measure has captured the effect of word 3 having a large number of senses. However, the entropy for word 3 is higher than the entropy for word 1. This is unexpected as most WSD systems would achieve greater accuracy for word 3 than word 1 due to its higher frequency baseline. This highlights the weakness of using entropy as a measure of the difficulty of a WSD task. The two factors entropy considers, frequency distribution and number of senses, are not weighted correctly. For WSD, the frequency of the major sense has a large effect on the accuracy. Its effect is much greater than the number of senses a word possesses. Therefore, entropy may be unable to give a good measure of the difficulty of the WSD task. This is investigated in section 6.9

The Cost/Error Matrix

It has now been shown that kappa and entropy incorporate a "difficulty of task" factor into their evaluation metric. Considering the difficulty of the task is only useful if the task used for evaluation is not the same for all systems.

The metric proposed in [Resnik and Yarowsky, 1997] incorporates a different factor into WSD evaluation. This factor is important even if the task is the same for all systems. The factor considered by Resnik and Yarowsky is a cost matrix. The cost matrix does not consider a system's choice of sense to be either right or wrong. A misclassification across broad sense distinctions is assigned a greater penalty than a misclassification between two very similar senses. This is enabled through the cost matrix.

A metric to evaluate the potential damage a system's misclassification may cause seems appropriate for WSD. It is often not necessary to determine the correct sense of a word for real NLP tasks to be successful. The correct broad meaning, which could be achieved from other senses, is often sufficient. How to derive a cost matrix to encode the appropriate penalties is the challenge required in order to use this metric.

Resnik and Yarowsky propose three methods for calculating a cost matrix. The first method proposed uses a *semantic distance* measure [4][Resnik, 1995b], [Richardson *et al.*, 1994]. Calculation of a semantic distance requires a hierarchical structure within the lexicon. It is this hierarchy which enables the calculation of a distance representing the similarly of two senses. It would not be possible to calculate a cost matrix using semantic distance for a lexicon with no hierarchical structure.

The second method proposed uses a *communicative distance* measure. This measure is based on psycholinguistic work and studies how closely related humans perceive different senses [Miller and Charles, 1991]. This is a labourious process and would be difficult to achieve on a large scale.

[4]Semantic distance is discussed in chapter 3

The final method proposed is the most promising, as it relates the error conceived in WSD to a real NLP task. Consider machine translation as an example of a real NLP task. The cost of misclassification relates to the chances of the chosen sense being mapped to the incorrect word in the target language. Using bi-lingual dictionaries, it would be possible to obtain large scale cost matrices for machine translation. Whilst this method is promising for machine translation, it is less easy to conceive how equivalent cost matrices could be developed for other NLP tasks. Also the benefit of the cost matrix must not be over estimated. The method is only valuable for evaluating WSD systems. It does not enable a measure to be determined to quantify the amount WSD improves the accuracy of a specific NLP task.

2.6 SENSEVAL

Many of the problems involved with the evaluation of WSD systems which were considered in the previous section were identified in [Resnik and Yarowsky, 1997]. These problems were discussed at the 1997 SIGLEX conference and as a consequence Adam Kilgarriff took up the challenge of developing a solution. The first major step towards developing an established solution took place in September 1998 with the pilot SENSEVAL evaluation. This was the first ever MUC style [Kaufmann, 1995] conference organized specifically for WSD. An outline of the format of the evaluation, and some of the considerations made in establishing SENSEVAL is now given. An equivalent evaluation was run at the same time for French and Italian called Romanseval. However, this account will concentrate on the English evaluation. A more detailed account is given in [Kilgarriff, 1998b] and [Kilgarriff, 1998a].

Kilgarriff's principle objective was to ensure that all types of system were able to compete. The primary challenge facing Kilgarriff was to find a corpus of manually sense tagged text which could be used for the evaluation. There were three choices: SEMCOR, DSO and HECTOR. SEMCOR and DSO are both readily available

and tagged with WordNet senses. WordNet does not supply complete dictionary definitions with all their senses. Therefore, both of these corpuses would make it difficult for systems which rely on dictionary definitions to compete. Also, as SEMCOR adopts a textual approach, there are less than 100 senses which occur more than 500 times. The DSO corpus does adopt a lexical approach, and there are many tagged instances of 191 commonly used nouns and verbs. However the tagging accuracy of the DSO corpus is not perceived to be high enough for an evaluation.

2.6.1 The HECTOR Data

Fortunately the HECTOR corpus provided a better solution. The HECTOR project is detailed in [Atkins, 1993] and contains 200,000 tagged sentences taken from the British National Corpus. HECTOR also adopts a lexical approach, and there are more than 300 words which possess over 100 tagged instances. Moreover the HECTOR data is not readily available, so no systems were at an advantage by using the data before the evaluation. The quality of the corpus was tested by re-sense tagging it in order to find the ITA. A figure of 96.5% ITA achieved was considered high enough to warrant the corpus being considered as a gold standard.

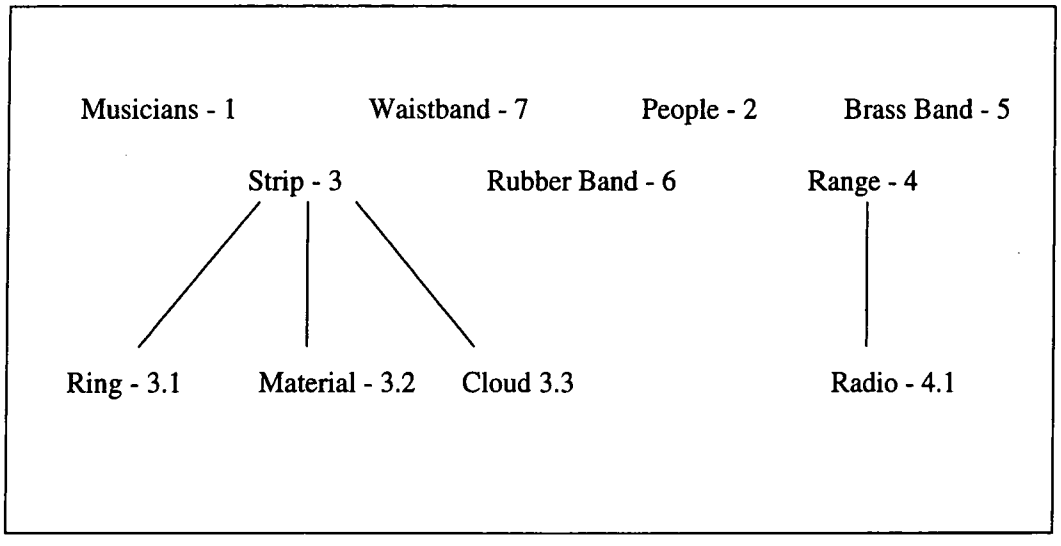
Within the HECTOR lexicon the division of a word into its component senses forms a shallow hierarchy. Senses which are considered to be completely separate are aligned at the top of the hierarchy. Senses which are semantically similar are all grouped as children of a broader more vague sense. The broader sense aims to encapsulate the possible meanings of all of its children. Other senses may be grouped together if they are semantically the same, but have syntactic variations. The word *band* is used as an example to highlight the format of the HECTOR hierarchy. Some of the senses identified for *band* in HECTOR are given in table 2.3.

Table 2.3 highlights many characteristics of the HECTOR lexicon. Two instances of the HECTOR hierarchy are shown. In both cases the parent sense (senses 3 and 4) has a broad definition so that it encapsulates all possible varia-

Table 2.3: Definitions for some of the senses for *band* identified in the HECTOR lexicon

Sense Number	Definition
1	A group of musicians.
2	A group of people - "A select band of athletes"
3	A strip of material
3.1	A gold wedding ring
3.2	Strip of an object different from main colour of material
3.3	Area in the form of a long flat strip - band of cloud
4	A range of values within a series
4.1	A range of frequencies or wave lengths particularly radio
5	Brass band
6	Rubber band
7	Waistband

Figure 2.1: Diagram showing the structure of the HECTOR hierarchy using the senses for *band* identified in table 2.3



tions of the sense. The main variations are then given as sub-senses. The sub-senses themselves can be quite different from each other, as is the case for the sub-senses of 3. A human should have no difficulty in determining which sub-sense of 3 was being referred to in any context. The sub-senses identify specific domains in which the more general sense is frequently used. Sense 4 is almost always referred to within the context of radio frequencies, and thus a sub-sense (4.1) is assigned to this specific domain. As a result, the more general sense 4 is infrequently tagged in a corpus. The more general sense will be used if a sense is used outside a domain in which a sub-sense has been identified. For example, sense 4 would be chosen in the following context - "Profit margins were in the 20% - 30% band". The *r*ange of values topic domain does occur frequently enough to warrant its own sub-sense.

The development of the HECTOR lexicon has been corpus driven, and this is reflected by the choices made in determining the structure of the hierarchy. In theory, the music band sense (1) should be a sub-sense of the more general group of people sense (2). After all, a music band is a group of people who play music together. The choice to consider these two senses are completely separate is primarily because of the large number of occurences of *band* referring to the music sense.

The example also highlights the way in which HECTOR deals with collocations and idioms. *Brass Band* and *Rubber Band* are both considered separate from any of the other senses identified. This choice may be because senses which have a definite collocate are easy to identify in free text. However, the hierarchy would better represent the semantic similarity between senses if *Brass Band* was a sub-sense of the music band sense (1) and *Rubber Band* was a sub-sense of the strip of material sense (3). Finally the example shows how a separate word such as *waistband* maybe included as a possible sense. It is trivial to choose a separate word as the correct sense in a text. Therefore, the inclusion of such senses distorts metrics used to quantify the difficulty of the task. The metrics affected include the number of possible senses for a word, and the baseline accuracy achieved by considering all senses with equal probability.

2.6.2 Evaluation Mechanism for SENSEVAL

The evaluation mechanism adopted for SENSEVAL is based on a probabilistic proposal by [Melamed and Resnik, 2000]. Other issues concerning the evaluation mechanism can be found in [Kilgarrif and Rosenzweig, 2000]. A summary and discussion of the evaluation mechanism is now given.

The evaluation mechanism enables both stochastic and deterministic systems to be evaluated. This is done by the system scoring the probability it assigned to the correct sense of the ambiguous word. A deterministic system will score either 1 or 0 in all instances. Whether this mechanism enables both systems to be evaluated fairly was discussed in section 2.5.2.

Fine grained and course grained results are both obtained. For the fine grained results, the same sense must be chosen as the tagged sense in the test data. For the course grained results, any sub-sense which belongs to the same main sense may be chosen. *Band* is used as an example to highlight the difference between coarse and fine grained results. If the correct sense is the cloud sense of *band* (3.3) then for the fine grained results the cloud sense must be chosen. However, for the coarse grained results senses 3, 3.1, 3.2 and 3.3 will score equally well if chosen. A mixed grain metric also exists which is a mixture of the fine and coarse results. Full credit is given for choosing senses 3 or 3.3 and partial credit is given for senses 3.1 or 3.2.

The coarse grained results may seem a more appropriate metric, because in many instances the sub-senses are very closely related. Few NLP tasks would need to distinguish between senses to the level required by the fine grained metric. However, the coarse grained metric relies on the HECTOR hierarchy. As was shown earlier, the structure of the HECTOR hierarchy is corpus driven and can cause semantically related senses to be far apart in the hierarchy. As a result, the credibility of the coarse grained results is reduced.

For all metrics, a precision and recall figure is given. This is done to encourage systems to participate which are unable to disambiguate all types of ambiguity. The

precision figure gives the system's accuracy out of all the instances it attempted. The recall figure gives the system's accuracy out of all possible instances. The difference between the precision and recall figures gives an indication of how many instances each system attempted.

$$Precision = \frac{Number\ of\ correct\ answers}{Total\ number\ of\ test\ sentences} \tag{2.3}$$

$$Recall = \frac{Number\ of\ correct\ answers}{Number\ of\ attempted\ test\ sentences} \tag{2.4}$$

2.6.3 Format of SENSEVAL Evaluation

The evaluation comprised of the competitors being assigned a manually sense tagged training data and extensive dictionary definitions for a set of 31 words. A further four words were assigned for which the dictionary definitions were given but there was no training data. For five of the words the POS of the correct sense was not known. The competitors had approximately two months to work with the training data and dictionary definitions to enable their system to disambiguate HECTOR senses. After that time all systems were frozen and test data was released for the same set of words. The results from all the systems were submitted and evaluated. An analysis of the results is given in chapter 7.

2.7 Criteria for Success

Now that some of the problems associated with WSD have been defined, the criteria for success for this work can be stated. The overall aim of this work is to produce a state-of-the-art WSD system. In order to be able to determine whether this has been achieved, it is necessary to set specific goals which relate to the seven NLE goals discussed in section 1.2.2. Three of the NLE goals are considered most relevant to this particular task. Showing that the system developed achieves these three goals is the primary criteria for success for this thesis. Highlighting three of the NLE goals does not imply that the remaining four goals will not be achieved.

The three NLE goals, and their criteria for success are now detailed:

2.7.1 Usability

The usability criterion refers to producing something which is actually desired by an end-user. For WSD, the end user is a developer of a NLP system. Accuracy is the most important criterion for a NLP developer wishing to incorporate a WSD module. Achieving 100% disambiguation accuracy would be an unrealistic criterion for success. Therefore, in order to fulfil this criterion, it must be shown that the system is able to achieve at least the same level of accuracy as other disambiguation systems performing the same task.

2.7.2 Flexibility

The ability to use a WSD system in different domains and for different tasks is a measure of its flexibility. Section 2.3.1 examined the problems associated with different systems working in different domains. The ability of the WSD system to be independent of any one domain is another criterion for the success of this system. This will enable it to be useful for many NLP tasks. The system develops a learning algorithm and requires sense tagged training data. Therefore, for this system two requirements are imposed in order to achieve domain independence.

- If mappings are available, a system trained on one lexicon can be applied to a separate lexicon without re-training.
- The learning algorithm developed must not be dependent on any one particular lexicon.

2.7.3 Scale

NLP systems should be able to process free text, which has not been written specifically for the NLP task. Therefore, there exist two dimensions to the scale

criterion relevant to WSD.

- The disambiguation system must be able to disambiguate text regardless of the number of sentences, or the length of any of the sentences in the text.
- The system must have wide coverage. The usability and flexibility criteria must be fulfilled for all words of all parts-of-speech which are found in a lexicon.

2.8 Summary

To summarise, the major problems associated with WSD are as follows:

- Large scale lexicons which are used as sense inventories assign a large number of senses to a word.
- Many senses are very similar in meaning.
- Sense distinctions vary greatly between lexicons, so comparisons can not be made, and the level of complexity of the task can not be quantified.
- To test a system a corpus of manually sense tagged text is required. This corpus is both labourious to produce and will not be completely accurate.
- Even with an accurate corpus there exists no metric which is able to evaluate all systems equally.

The chapter then examined the framework of the SENSEVAL evaluation which aims to overcome many of the problems identified above. Finally the criteria for success for this work were detailed relating them to the goals of NLE identified in chapter 1.

Chapter 3

Related Work

Chapter 2 details many of the problems which a developer of a WSD system must consider. This chapter examines what has already been achieved within the field. The chapter focusses on the many different sources of information which have been used to help resolve WSD. This chapter discusses and provides criticism for each of these knowledge sources and examines ways in which they have been used by system developers. Less emphasis is placed on reporting the results achieved by other systems. WSD Evaluation performed on different data sets is difficult to compare as was discussed in chapter 2. Emphasising the results achieved by different systems may imply that one system is better than another. However, this implication could be very misleading.

Before any of the modern day state-of-the art systems are considered, a brief discussion is given of how this field has developed.

3.1 A Brief Look Back

Early work investigating WSD began in the 1950's. The research was inspired by automatic machine translation systems which identified the need for accurate sense discrimination. This work identified many of the principles behind which modern day techniques are based.

The importance of local context to help humans resolve ambiguity is discussed in [Weaver, 1955]. Subsequent work found that humans could normally resolve sense ambiguity if they were given two contextual words either side of the ambiguous word. Local context was therefore considered important for automatic systems. Weaver also identified a need to use statistical techniques due to the uneven nature of a sense's frequency distribution. Grammatical structure was also considered by Reifler in 1955. Reifler shows how the ambiguity of *keep* can be resolved by examining the grammatical type of its object:

- *He kept a record* - noun phrase.
- *He kept calm* - adjectival phrase.
- *He kept eating* - gerund.

However, in the 1950's machine translation was only being developed for specific domains, in which much ambiguity is already resolved. For example, in a mathematics domain, *root* can only refer to the square root, and not to the part of a plant. Moreover, these techniques remained predominately theoretical ideas. There was not the lexical or computational resources available to enable them to be expanded. However, many of these theoretical ideas have been adopted in more modern day approaches. For example, [Kilgarriff, 1997a] proposes a foreground and background lexicon for IE. The foreground lexicon is specific to a particular domain to aid the WSD process. The remainder of this section examines a variety of modern day approaches to WSD and shows how the techniques discussed in the 1950's are still relevant today.

3.2 Dictionary definitions

Using sense definitions given in a dictionary is a convenient way to resolve ambiguity. This approach has many advantages. As Machine readable dictionaries are widely used in many domains they are readily available. Moreover, manually

sense tagged training data is not required. The value of this characteristic is shown in chapter 2, which highlights the difficulties involved in creating manually sense tagged data. The method used to develop a system for one dictionary may also be able to be applied to a different dictionary. Potentially this enables the system to operate in several domains and increases the portability.

The use of dictionary definitions was inspired by [Lesk, 1986], who used the Oxford Advanced Learners' Dictionary (OALD) [Hornby, 1963] dictionary for the definitions of senses. Ambiguity is resolved by examining the number of overlaps between the ambiguous word and the context words. An overlap is the number of words a sense's definition has in common with the definition of a context word. Lesk uses *pine cone* as his example. The correct senses of both *pine* and *cone* can be determined as *tree* is common to both definitions.

[Véronis and Ide, 1995] show that Lesk's method can be adopted to correctly resolve the ambiguity of *pen* if *sheep* is a context word. Véronis and Ide show this using the *Collins English Dictionary* (CED) [Hanks, 1979], instead of the OALD dictionary as used by Lesk. The use of a different dictionary highlights the potential portability characteristic of the dictionary definition approach. However, Véronis and Ide proceed by showing the fragility of Lesk's method due to its reliance on the particular wording of the dictionary. If either *chicken* or *goat* are the context words instead of *sheep*, the ambiguity of *pen* is unable to be correctly resolved. The definitions of *chicken* and *goat* contain no words which are also contained in the definition of *pen*. With the writing utensil sense of *pen* a converse problem is observed. If *page* is a context word then seven senses of *pen* have the same number of overlaps. There is therefore no way to determine which of these is the correct sense.

Lesk's work was built upon by [Cowie *et al.*, 1992] who determined a way to simultaneously resolve the ambiguity for all words within a context window. Cowie et. al. applied Simulated Annealing techniques to the disambiguation problem. Simulated Annealing is a search mechanism which has been successfully applied in many other aspects of AI such as the travelling salesman problem. Simulated

annealing requires a function E which depends upon the particular configuration of the system. The aim is to minimize E. A new configuration is chosen randomly from a set starting point. If E is reduced, the new configuration is chosen. If E increases, the new configuration may still be chosen enabling the algorithm to move up hill and thus escape local minima. A solution is derived when no more improvements can be found.

Within the context of WSD, the configurations are the sense choices for each word in the context window. In the work of Cowie et. al., E is calculated in terms of the number of word overlaps in the definitions of the senses of a particular configuration. The lower the value of E, the more overlaps there are, and hence the configuration is more likely to have the correct senses. This search algorithm proved very successful when used with the overlap function for dictionary definitions. However, simulated annealing is not restricted to use with the dictionary definition approach. Any other disambiguation approaches which are able to assign a score to a configuration of sense choices are able to adopt simulated annealing techniques.

Cowie et. al.'s system improved on Lesk's work by enabling larger context windows. Lesk limited his context window to ten words, Cowie et. al. used a whole sentence as context. As a consequence, inconsistencies in the dictionary definitions were smoothed over a large data set, thus reducing the effect on results. However, Lesk's and Cowie et. al.'s system did not fully realise its potential capabilities. The reason for this is that the length of the dictionary definitions was not considered. If a sense has a long definition, the likelihood that some words in the definition will overlap is increased. Therefore, senses with short definitions and senses which are defined in terms of their synonyms are penalised.

Cowie et. al.'s work is very important for the simulated annealing search algorithm it developed. However, the information extracted from the machine readable dictionary was similar to that which Lesk had used. This richness of the information extracted from the dictionary is improved in [Véronis and Ide, 1990] and is later presented in more detail in [Véronis and Ide, 1995]. Véronis and Ide identify

a dictionary as a highly connected network of words and concepts. By doing so, they are able to consider not only the words in a definition, but also the definitions of the words in a definition, and so on. Using the CED as their machine readable dictionary, Véronis and Ide produce a very large neural network. The nodes in the network represent concepts in the dictionary and the connections between them represent a semantic relationship between words. Once trained, this neural network can be used for WSD. Context words in a text will activate different senses of an ambiguous word based on their relationship in the dictionary. The sense with the highest activation will be chosen. Other connectionist approaches have been applied in different ways to the WSD problem in [Lyon, 1994], [k Kawamoto and Anderson, 1994], and [G.Cottrell, 1984].

Véronis and Ide provide an intriguing method to extract semantic information from a dictionary. Their approach is limited by their underlying assumption - *"there are significant semantic relations between a word and the words used to define them."* - [Véronis and Ide, 1995]. It is evident that the dictionaries do contain semantic information which is able to help disambiguation. However, there will always exist many instances where the dictionary information is insufficient. Dictionary definitions provide information concerning the meaning of a sense. These definitions do not contain information regarding the distinctions between different senses. An alternative approach to extracting semantic information from dictionary definitions is given in [Chodorow and Byrd, 1985].

This limitation of the dictionary definition approach is realised in [Wilks and Stevenson, 1997b] and [Wilks and Stevenson, 1998]. Their basic approach builds upon Cowie et. al.'s work by using simulated annealing with LDOCE dictionary definitions. Wilks and Stevenson approach the problem caused by a variation in dictionary definition lengths. The contribution a word makes to the overlap function is normalized by the number of words in the definition. Therefore, a word contained in a long definition will contribute less to the overlap function than a word contained in a short definition. This normalization produces a small improvement in accuracy.

3.3 Other Dictionary Information

All the systems examined in the previous section used the definitions in the dictionary as their knowledge source. This section discusses other information available in a dictionary which also may aid disambiguation.

In addition to the LDOCE dictionary definitions, Wilks and Stevenson combine two other knowledge sources from the LDOCE dictionary. LDOCE assigns **pragmatic codes** to all entries. These codes represent a subject category to which a word belongs. Primary and secondary codes exist producing a shallow hierarchy. Senses are chosen to achieve the greatest overlap of pragmatic codes within a text. The overlap is optimized over a paragraph of text as this knowledge source tries to capture general topic information.

LDOCE also facilitates the use of basic **selectional preferences** which Wilks and Stevenson also take advantage of. Each noun is assigned one of 35 semantic classes identified by LDOCE. Within the definition for each adjective, adverb and verb, information is given regarding the semantic classes each word is likely to modify or possess as arguments. This information enables the elimination of senses which do not fall into the required semantic class. [Morgan *et al.*, 1995] uses *"He drove the train"* as their example to demonstrate their use of selectional restrictions for disambiguation. The correct sense of the verb *drive* must take a vehicle as its object. Therefore, the *part of a wedding dress* and *sequence of events* senses of *train* can be disregarded as they are the wrong semantic class to be an object of *drive*. Equally the correct sense of *train* helps to resolve the ambiguity of *drive*. The ability to identify the grammatical links within a sentence is a pre-requisite for using selectional restrictions. Automatically identifying the grammatical links using a shallow parse will introduce some error into the system [Basili *et al.*, 1992]. Also in some cases, the correct sense may not belong to a semantic category allowable by the verb's selectional constraints. The example used to highlight this point is taken from [Wilks, 1978]. The verb *drink* constrains its subject to an animate object. However, in the sentence *"My car drinks gasoline"*, this selectional constraint is broken. It is for these reasons that Wilks chooses to use the selectional constraint

information to weight possible senses known as preference semantics [Wilks, 1968]. This is in contrast to earlier restrictional approaches [Katz and Fodor, 1964] which did not consider instances where the constraints could give misleading information.

The information from selectional constraints is still used in current research [Wilks and Stevenson, 1997a], [Harley and Glennon, 1997], [Hirst, 1994], [Hearst, 1991] and [Gomez, 1997]. Also selectional restrictions are in general only useful for ambiguity resolution between broad senses. Finely grained senses are likely to belong to, or be used with, the same semantic class.

Wilks and Stevenson's sense tagging system is promising for its ability to extract from the LDOCE dictionary as much information as possible which may help WSD. Their ability to perform disambiguation on a large scale has enabled their system to be integrated into the GATE architecture [Cunningham *et al.*, 1998] and [Stevenson *et al.*, 1998]. GATE - a General Architecture for Text Engineering provides the organisational pattern for various components and knowledge sources which constitute text processing [Cunningham *et al.*, 1996].

However, the sense tagging system uses training data in order to establish the best way to combine the weights from each of the knowledge sources. This requirement on training data loses one of the initial advantages of building on Cowie et. al.'s work. The system has also lost another initial advantage of the dictionary definition approach. The system is no longer able to be used with any machine readable dictionary, but is dependent on LDOCE. It is partly for these reasons that Wilks and Stevenson's system was unable to take part in the SENSEVAL evaluation [Wilks, 2000]. As a result, there exists no acceptable way to compare their system with other state-of-the-art systems.

Similar knowledge sources to those employed by Wilks and Stevenson are used in [Harley and Glennon, 1997]. Harley and Glennon use the CIDE dictionary as their knowledge base. The main advantage of this system is it is being closely developed with the lexicographic team responsible for the CIDE dictionary. This partnership enables them to have an input into which dictionary information is beneficial for disambiguation. Like LDOCE, the CIDE dictionary offers subject

Table 3.1: Table showing the words (given in bold) identified in the CIDE dictionary which help to distinguish between five senses of *bang*

Sense	Definition	Clue
bang 1 0	To make a sudden loud noise	someone banging **at/about**
bang 1 5	To make a sudden loud noise	**went** bang
bang 2 0	To accidentally hit something	banged his head **against/on**
bang 3 0	To have sex	banging **away**
bang 4 0	Exactly or directly	bang **in the middle**, **slap** bang

domain tags and selectional preference tags. As well as these two data sources, the CIDE dictionary also identifies clues which may appear in the context and help to determine the correct sense of a word. Table 3.1 is used to show the type of clues CIDE identifies using *bang* as its example. CIDE uses examples with these clues in to show where they must be positioned relative to the ambiguous word. This additional knowledge source makes CIDE an excellent resource for WSD. Unfortunately the CIDE dictionary is a commercial product and is still under development. Therefore, CIDE is not readily available. These two factors prevent it from being used more widely. Harley and Glennon's system uses additive weights to combine their different knowledge sources. These weights are manually set, based on a subjective opinion for the value of information from each knowledge source. Whilst the value for the weights may not be optimal, it does prevent their system requiring any form of training data.

3.4 Thesaurus

Unlike dictionaries, thesauri do provide information concerning the relationship between words and the similarity of senses. Therefore thesauri could provide more useful information to distinguish between senses than dictionaries. The most commonly used machine readable thesaurus for WSD is the *Roget's International Thesaurus* [Chapman, 1977]. The measurement of overlap using a thesaurus list of words rather than a dictionary definition may seem appropriate for WSD. By the

very nature of a thesaurus, the words listed together are similar, and therefore may provide useful contextual information. However, this approach using a thesaurus as a knowledge base is not well developed. Instead, the Roget's thesaurus has been used in a different way.

Within Roget's thesaurus, 1042 semantic categories have been identified. Each word in the thesaurus is assigned one, or several, of these semantic categories. Roget's thesaurus does not split a word into component senses. Individual senses broadly equate to a single semantic category assigned to a word. It was this information that was used by [Yarowsky, 1992] and [Gale *et al.*, 1992a] to resolve word ambiguity. These semantic categories are similar to the pragmatic codes found in LDOCE and the subject domain tags found in CIDE. However, Yarowsky introduces a novel way to utilise this resource, this method is described in a three step process.

- **Collect contexts for each Roget's category:** The first step is to collect a set of context words which frequently appear with a word belonging to a particular Roget's category. As the corpus used to collect these words is not tagged with their Roget's categories, some noise is introduced through polysemy.

- **Identify salient words:** It can be seen from the set created in step one, that many words will exist because they frequently occur. Step two involves identifying salient words in the set, which appear significantly more often in the context of one category than in the rest of the corpus. This is achieved by adopting probabilistic measures, and results in the identification of words which provide useful contextual information for that category. Table 3.2 highlights some salient words found for two Roget's categories.

- **Use salient words to resolve ambiguity in novel text:** If salient words identified in step two appear in the context, they provide evidence for the particular Roget's category for which they have been identified for. A large contextual window is used which consists of 50 words before and 50 words after the ambiguous word. In such a large context several salient words are

Table 3.2: Table given in (Yarowsky 1992) to show some of the salient words found to help identify a Roget's category.

Roget's Category	Salient Words
Animal/Insect	species, family, bird, fish, breed, egg, centimetre, animal, tail, wild, common, coat, female, inhabit, eat, nest
Tools/Machinery	tool, machine, engine, blade, cut, saw, lever, pump, device, gear, knife, wheel, shaft, wood, tooth, piston

likely to appear. The weights for all words found in the context are added together, and the category with the greatest sum is chosen.

Using topic information in the aforementioned way is a very promising approach. Other knowledge sources aim to find words which are similar or related to the ambiguous word. This is based on the assumption that similar words do provide useful clues. Yarowsky's approach tackles the problem more directly by identifying words which provide contextual clues. As table 3.2 shows, not all the identified salient words would be considered similar to words in a particular Roget's category. For example, *centimetre* is not similar to an *animal/insect*, but it is still able to provide a useful contextual clue to help determine the correct Roget category. Yarowsky found that this knowledge source performed most efficiently if a wide context window is used to capture the topic information. 50 words either side of the ambiguous word were considered. The same large context window was also adopted in [Rigau and Agirre, 1995] where the knowledge source developed also aims to capture the general topic information.

There are difficulties involved with using the Roget's categories, as is highlighted in [Ellman *et al.*, 2000]. Ellman et al report a degradation in performance when Roget's semantic categories are included as a knowledge source. The technique finds *"many spurious relations where words in the local context are interpreted ambiguously."*. Moreover, Yarowsky reports that for many words, some ambiguity still remains even if the correct Roget's category is chosen. Roget's semantic categories

can not be used to resolve these fine grained sense ambiguities.

The use of a large context window is typical of an approach which tries to encapsulate topic information. This type of topic information has proved to be beneficial for disambiguating nouns. Verbs and adjectives benefit more from local context. This approach proves to be particularly beneficial for nouns which are topic constrained. For example, the money sense of *interest* is predominantly used within a financial domain. However, other senses of *interest* are less constrained to a single topic and hence this approach is less reliable for resolving their ambiguity.

3.5 WordNet

This section examines a hierarchical lexicon which is a quite different resource from the machine readable dictionaries discussed above. This kind of hierarchy may be consistent with the method in which humans organise their mental lexicon. As a result, there is interest in hierarchical networks across many different fields. However, within NLP, the underlying purpose of a hierarchical network is similar to dictionary definitions. The purpose of a hierarchical network is to quantify the similarity between words. This is important for WSD as similar words are likely to appear in the same context. This section examines approaches to WSD which have used a hierarchical network resource.

The best known and most commonly used hierarchical network within NLP is WordNet. WordNet is a lexical database and has been a continuous research project at Princeton University since 1985. WordNet is dissimilar to a dictionary as lexical information is organised by semantic properties rather than spelling. Although later versions of WordNet do contain definitions for most words, these definitions are merely helpful extras rather than a principle component. The meaning of a concept is determined by its position in the hierarchy.

"The theory we were testing assumed that, if you got the pattern of semantic relations right, a definition could be inferred from that - it seemed redundant to

include definitions along with the network of semantic relations" [Fellbaum, 1997]

The building blocks for the WordNet lexicon is a synset. A synset is a group of words which express a single concept. If words can be interchanged in **some** contexts, they are identified as synonymous in WordNet. A stronger requirement for interchangeability within all contexts would be impractical and lead to very few synonyms. Polysemous words are defined by appearing in more than one synonym. Each synonym in which the word appears represents a concept which refers to a sense of the word. The strong use of synonyms in WordNet resembles the structure of a thesaurus. Indeed, WordNet can be used as a thesaurus to help the user find the correct word to express a concept. However, WordNet offers further information about the relationship between these synsets and this is one of its main advantages. This information is not available in a thesaurus.

The type of links found between synsets is dependent on the part-of-speech. The organization of nouns, verbs and adjectives within WordNet is now described.

3.5.1 Nouns

The synset groups for nouns are linked by hypernyms and hyponyms. A hypernym is a generalisation of a concept. For example, a *citrus fruit* is a hypernym of an *orange.* A hyponym represents a specialization and is the inverse of a hypernym. Continuing the example, a *citrus fruit* is a hyponym of an *edible fruit.* Hyponyms can be considered as an "is a kind of" link - *an orange "is a kind of" citrus fruit.*

WordNet categories synsets based on lexical rather than discourse semantics. This causes the **Tennis Problem** [Fellbaum, 1996]. All of the concepts associated with a game of tennis are distributed across the WordNet Hierarchy. A tennis player is a hyponym of *person.* The racket, balls, net etc. are hyponyms of *artifact.* The tennis shots are hyponyms of *actions* and tennis itself is a hyponym of *activity.* The distributed nature of elements in WordNet related to tennis highlights the difficulty of using WordNet to extract topic information from a text. This is shown in figure 3.1. The tennis problem highlights a problem with using WordNet for

WSD. Developers who use WordNet for WSD often make the assumption that the correct sense of a word is the one which is close in the WordNet hierarchy to other concepts in the sentence. The tennis problems demonstrates that a word which provides useful information in the context of an ambiguous word may not appear in the same section of the WordNet hierarchy.

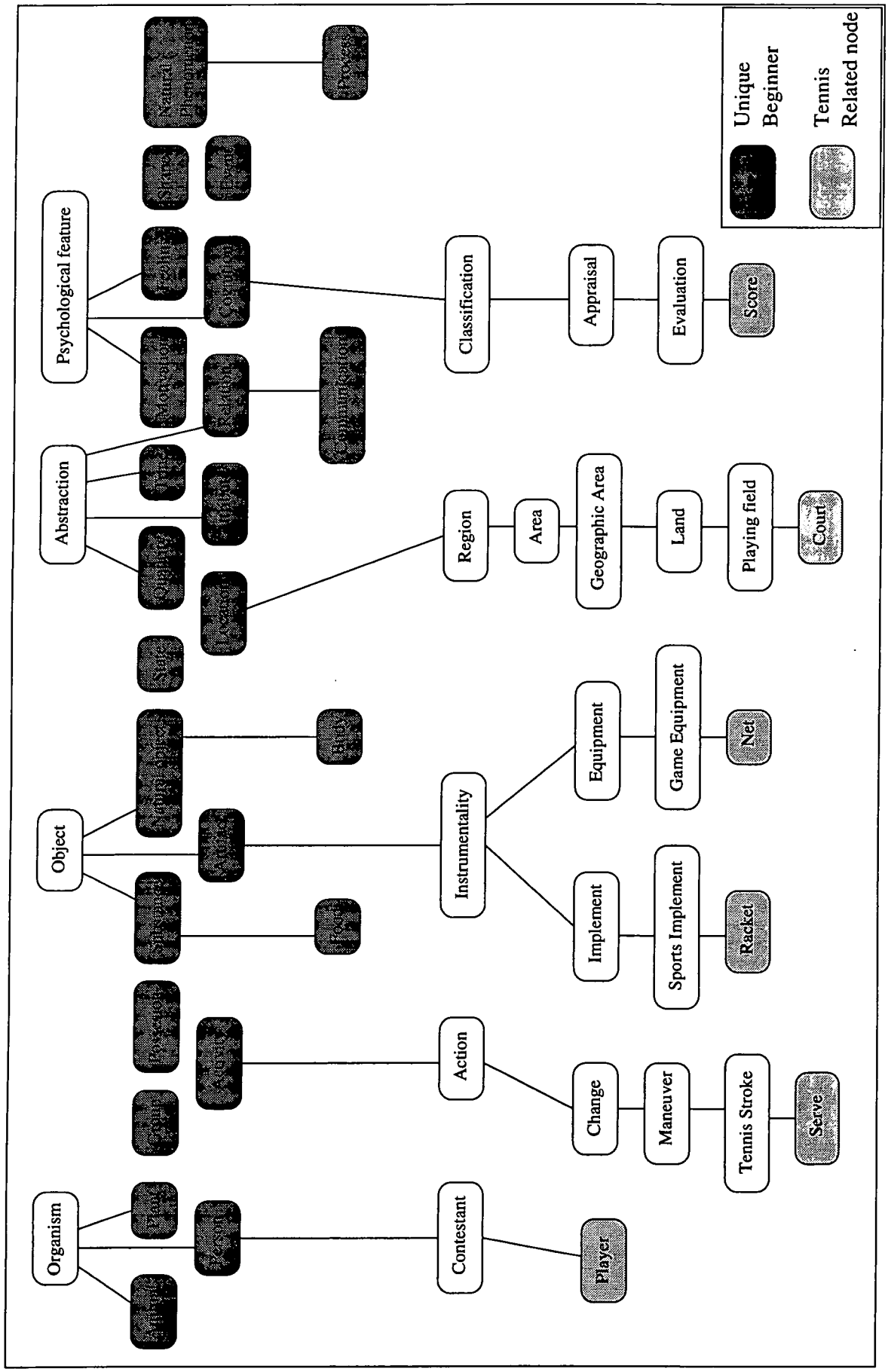
Furthermore, the purpose of the WordNet hierarchy is to distinguish among hyponyms, rather than fully represent all the features of a particular concept. Using the earlier example, of *orange*, the hierarchy shows that it is a type of citrus fruit. However, there are no links to other concepts which define the characteristics of *orange*. For example, orange could be defined as a specialization of *things that roll*, *things that are orange*, *things found in a supermarket* etc.

Synsets in WordNet do not lead up to a single root concept serving as the hypernym of all nouns. Instead, 25 *unique beginners* have been identified, which form the basis for the top level of the WordNet hierarchy. The *unique beginners* have been chosen in order that each category covers a distinct lexical domain. Some overlapping between categories is unavoidable. Analysis of the types of nouns that an adjective could modify was carried out. This analysis was influential for the selection process of the unique beginners. The high level structure for WordNet showing all the unique beginners is shown in figure 3.1

3.5.2 Verbs

Like nouns, the synonyms of verbs are grouped together into synsets. There are very few true verb synonyms where one verb can always replace another verb in a text. Due to the wide variety of contexts in which verbs can be used, some verbs will be semantically similar in some contexts, but dissimilar in other contexts. For example, in most contexts *rise* and *fall* have close synonyms *ascend* and *descend* respectively. However, if the subject is the temperature or stock market prices then the synonyms can not be interchanged. Difficulties arise in deciding which verbs can be grouped as synonyms and which can not be grouped.

Figure 3.1: Top level hierarchy of nouns in WordNet



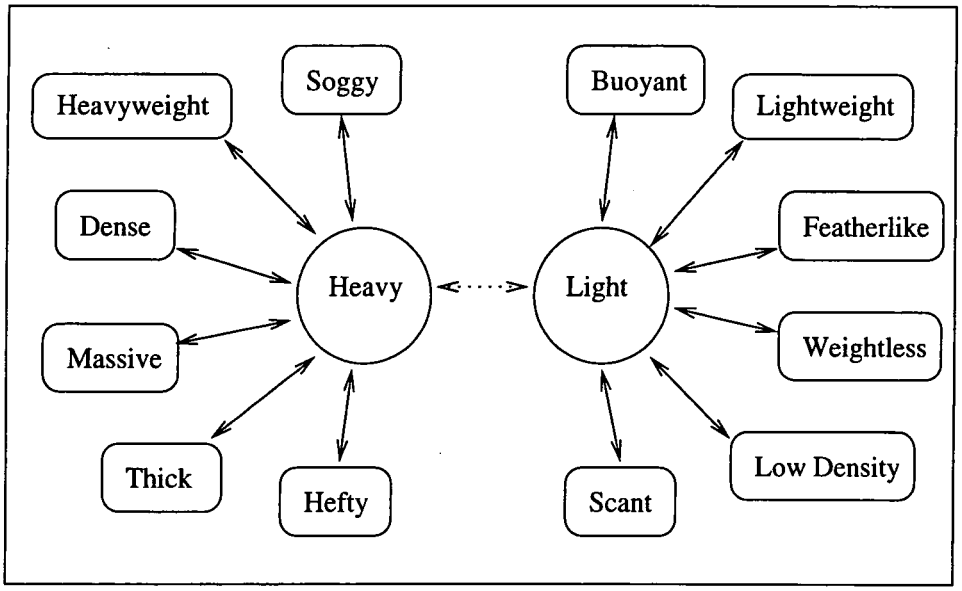
Moreover, as with nouns, the verb hierarchy contains 14 unique beginners which categorize the verbs into sections. Compared with noun categorization, verb categorization causes more difficulties as many verbs cross categories. For example, the verb *roar* in "The bike roared passed", describes both a motion and a sound. Hyponym links connect synsets in the hierarchy as they do for nouns. A hyponym link for verbs is equivalent to saying that *To V1 is V2* where V1 is a hyponym of V2. For example, *To sing is to perform.*

3.5.3 Adjectives

There exists no hierarchical structure available to organize adjectives, unlike for nouns and verbs where hierarchies do exist. In general it is meaningless to suggest that one adjective "is a kind of" another adjective. Adjectives fall roughly into two categories - **descriptive** and **relational.**

Adjectives which fall into the descriptive category constitute the larger proportion. This type of adjective are usually thought of as "common" adjectives such as *cold, heavy, hard* and *tall.* Descriptive adjectives refer to an attribute belonging to the noun which the adjective describes. For example *cold* modifies the temperature attribute of a noun. Therefore, a temperature attribute must belong to all nouns which *cold* is used to describe. Many descriptive adjectives possess an antonym which modifies the same attribute in the opposite way. The antonym of *cold* is *hot.* There also exists relationships between semantically similar adjectives that modify the same attribute in the same way, but to different extents. For example *cold* is semantically similar to *bitter, chilled, parky, frosty, crisp* and *raw.* It is these relationships which enable WordNet to organize the descriptive adjectives. A synset is defined as a group of semantically similar adjectives which modify an attribute in the same direction. One of the members of each synset is linked to their direct antonym. This antonym is a member of a synset of adjectives modifying the same attribute in the opposite direction. The adjective structure for words which modify the *weight* attribute are shown in figure 3.2.

Figure 3.2: Diagram showing the WordNet structure for descriptive adjectives



3.6 Approaches using WordNet

Now that the WordNet lexicon has been described, approaches which have used this resource for WSD can now be discussed. The approaches described aim to give a broad representation of different methods adopted for using the WordNet resource.

3.6.1 Overcoming Data Sparseness

The lack of adequate training data for WSD is a common problem. The system described in [Leacock and Chodorow, 1998] uses WordNet to identify text similar to text found in sense tagged training data. Leacock and Chodorow's work is based on the assumption that if the context is similar to the training text, it follows that the referred sense is the same in both texts. For example, if *cricket* provides contextual information to resolve the ambiguity of *play*, then similar ambiguous words to *cricket*, such as *rugby*, *football*, *tennis etc* will also provide evidence for the same sense of *play*. This approach provides contextual information for many words that have not appeared in the training data. Greater use is made of each sentence in the training data which helps to overcome the problem of data sparseness.

Leacock and Chodorow use the distance between two nodes in the WordNet hierarchy as their measure of semantic similarity. They calculate the path length between two nodes a and b as:

$$SIM_{ab} = \max[-log_2 \frac{Np}{2D}] \tag{3.1}$$

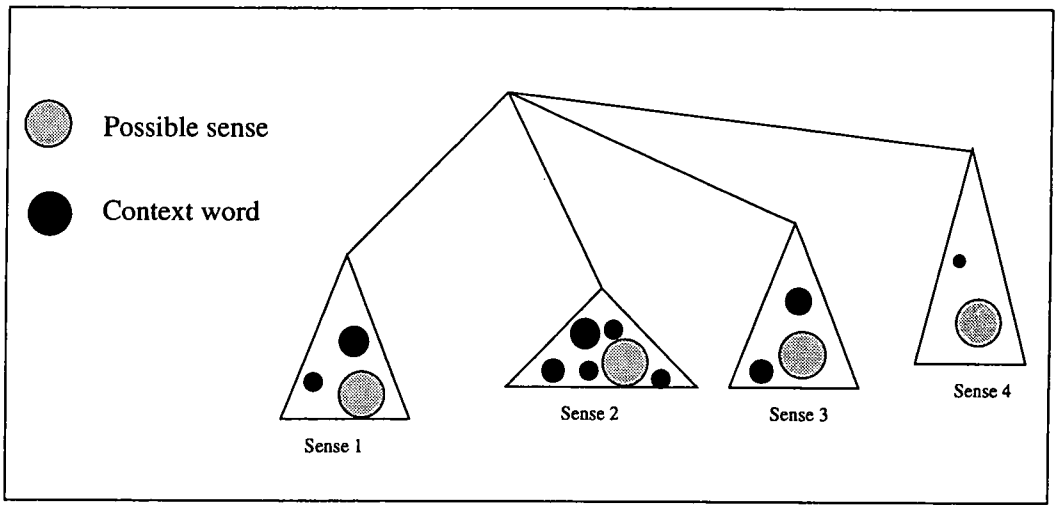
where Np is the number of nodes in path p from a to b and D is the maximum depth of the path in the hierarchy. Depth is considered as two concepts linked low down in the hierarchy are semantically more related than two concepts linked at the top. The measure uses the number of nodes in the path rather than the number of hypernym/hyponym links in the path, so that a score for two words in the same synset can be calculated.

3.6.2 Semantic Distance

The next method considered uses a measure of semantic relatedness in a different way to Leacock and Chodorow to resolve WSD. The difference stems from the initial assumption made. The assumption made in [Agirre and Rigau, 1995], [Agirre and Rigau, 1996] is that semantically related words appear together in the same text. Therefore, Agirre and Rigau choose senses which contribute to a high **Contextual Density** in one section of the WordNet's hierarchy. The WordNet hierarchy is sectioned so each possible sense of an ambiguous word is assigned its own section. The contextual density for each sense is calculated. This calculation is based on the number of contextual words found in a section and the overall size of the section. The sense which belongs to the section with the highest contextual density is selected. This is shown in figure 3.3

A similar approach is presented in [Sussna, 1993]. Sussna uses the same assumption made by Agirre and Rigau - that semantically related words appear together in the same text. A formula is determined to calculate a semantic distance measure between two nodes in WordNet. Unlike Leacock and Chodorow, this formula considers the number of hyponyms leaving a node as well as the depth of a node in

Figure 3.3: Diagram showing the use of Contextual Density for sense discrimination. Sense 2 is chosen as it has the highest Contextual Density



WordNet. This is performed in order to compensate for the imbalance present in the WordNet hierarchy. Much greater depth exists in some areas of the WordNet hierarchy than others. There is a greater likelihood of a node with many hyponyms representing a general concept regardless of it being positioned low down in WordNet.

A different method for overcoming the unbalanced nature of WordNet is presented in [Resnik, 1995a]. The similarity of two concepts, a and b in the WordNet hierarchy is determined by the concept which subsumes both a and b. If the subsuming concept occurs frequently, it follows that both concepts are less semantically related. The frequency information is determined from training data. The frequency of each concept found in the text and all of its hypernyms are incremented by one. Resnik uses the following formula to determine his measure of semantic similarity:

$$SIM_{ab} = \max[-log_2(Pr(c))] \tag{3.2}$$

where $Pr(c)$ is the probability of a concept c which subsumes both senses a and b. Resnik reports that his measure for semantic distance is more closely related to a human's perception of similar words rather than edge counting measures [Resnik,

1995b]. However, this does not imply that Resnik's algorithm is a better measure for WSD. The approach adopted by Agirre, Rigau and Sussna's requires no training data proving an advantage over the measure used by Resnik.

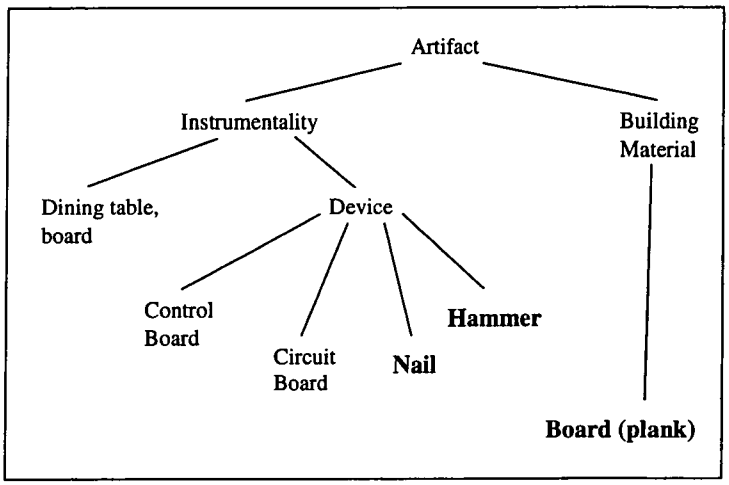
All these methods suffer from the **Tennis Problem** described in section 3.5. Many words provide strong contextual clues, which humans would consider related, but do not appear in the same part of the WordNet hierarchy. For example, the semantic similarity approach works well to aid the disambiguation of *pitcher* in the sentence: *"The baseball pitcher's injury would mean that he would miss the rest of the season"*. In WordNet the thrower sense of pitcher is a specialization of baseball. Therefore, baseball subsumes both concepts and each concept would be considered very similar. However, only the root node subsumes *pitcher* and either *injury* or *season* in WordNet. Therefore, if *baseball* was removed from the sentence, it would become more difficult to disambiguate *pitcher* correctly using a semantic similarity measure. A good disambiguation system should still be able to find the correct sense of *pitcher* using the contextual words *injury* and *season* as contextual clues.

Even if the algorithm correctly determines the most similar sense to the contextual words, this does not guarantee that the most similar sense is the correct one. Consider the sentence *"The pitcher called for a glass of water"*. In WordNet, *glass* and the "vessel" sense of *pitcher* are both subsumed by *container* and hence would give misleading contextual information. [Karov and Edelman, 1996] use *hospital* and *doctor* as their example to highlight a similar problem. These words are assigned a very low semantic similarity in WordNet as the former is a type of building and the latter a type of professional. Figure 3.4 highlights another example of this concept. *Hammer* and *nail* are most closely related to the circuit and control sense of *board* rather than the lumber sense.

WordNet categorizes words based on their lexical relatedness. This is one of the difficulties of using WordNet for WSD and is highlighted by the tennis problem. A measure of relatedness in terms of topic information may be more relevant for WSD.

Applying a semantic distance measure to WSD causes extreme difficultly in

Figure 3.4: Diagram showing that in WordNet the *board (plank)* sense is not the nearest to either *nail* or *hammer*.



using words with different parts of speech as context. In WordNet the noun and verb hierarchies are completely separate. Therefore, using the subsuming or node counting approach, no way exists of obtaining a similarity measure between concepts with different POS. This often means that the algorithm tries only to disambiguate nouns, using only nouns as context [Resnik, 1995a]. In the sentence *"The pitcher throws very fast"* the algorithm would be unable to use *throws* as a contextual clue to help disambiguate *pitcher*.

To summarize, WordNet is a very useful resource for NLP systems. Within the smaller domain of WSD there are also many ways in which WordNet can be usefully applied. However, there is a danger of expecting WordNet to accomplish more than its capability with respect to WSD. When using WordNet for WSD one must always remember what the developers of WordNet set out to achieve and work with those limitations.

3.7 Corpus Based Methods

The recent availability of data which has been manually sense tagged has been of enormous benefit to WSD. Manually sense tagged data and the availability of powerful computers have enabled many avenues of research to become practically

viable, on a large scale. For example, the Machine Translation system developed by Masterman in 1957 was unable to be tested on a large scale due to a lack of resources and processing power. Some approaches which have exploited the resources presently available will now be discussed.

An exemplar-based approach is given by [Hwee Tou Ng and Hian Beng Lee, 1996]. The principle behind Ng and Lee's method is similar to Leacock and Chodorow's, discussed in section 3.6. Both systems determine a similarity measure between sentences. Each sentence in a test set is compared with the sentences in a training set. The training sentence most similar to the test sentence is chosen. The tagged sense in the training sentence is then assigned to the ambiguous word in the test sentence. Unlike Leacock and Chodorow, the training data is not only used as a set of example sentences by Ng and Lee, but also to learn their similarity measure. The similarity measure is based on a set of features which have been identified as aiding the disambiguation process. These features are: part-of-speech of neighbouring words, morphological form, set of surrounding words and the verb-object syntactic relationship. During training, the system learns the effect different values of these features has on the sense of a particular ambiguous word. By doing so, distance measures can be obtained between different values of each feature. The importance of each feature for disambiguation can also be determined. The distance measures learnt are the basis of the similarity measure between sentences.

Ng and Lee's method is not dependent on any one particular lexicon and is a major advantage of this approach. This feature enables the algorithm to be portable and used in many domains. Moreover, Ng and Lee report that the method achieves high accuracy when tested on the ambiguous word *interest*. However, scalability is the main drawback of this system. 1769 sentences were used to train the system to disambiguate the word *interest*. It would be impractical to produce a corpus with this number of training examples for all ambiguous words. The system performed less well when tested on the common ambiguous words found in SemCor, this highlights the problem of the scalability feature of this approach.

The problem of needing a large number of training sentences for an exemplar

approach is considered in [Fujii *et al.*, 1996] and [Fujii *et al.*, 1988]. Examining ten Japanese verbs, it was found that 100 example sentences were needed for each verb. The manual resources required to sense tag this quantity of sentences for a large quantity of verbs was far too big. In addition, it was computationally expensive to compare each test sentence with such a quantity of example sentences. Fujii et. al. proposed a solution using selective sampling to identify the most informative sentences which aid example selection. The verb sense disambiguation system which uses the example sentences is described in [Fujii, 1998]. If the sentences have been selectively sampled prior to being used in the verb sense disambiguation system, the system requires only one third the number of training sentences.

The sense tagged corpus has also enabled a range of probabilistic techniques to be employed for disambiguation [Key-yih Su *et al.*, 1992] and [Chang *et al.*, 1992]. Identifying a range of informative features for ambiguity resolution is the fundamental principle underlying this approach. The probability of each sense occuring, given the presence of a particular feature, is then calculated. The probabilities are calculated based on Bayes theorem:

$$P(s|x) = \frac{Freq(s \cup x)}{Freq(x)} \tag{3.3}$$

where s is a sense of an ambiguous word and x is a feature.

Bayes theory is used in [Brown *et al.*, 1991] where collocations are adopted for the features. Similar probabilistic techniques are present in [Yarowsky, 1992] where the features used are based on the Roget's semantic categories.

Producing probabilistic models becomes more difficult if many features are considered. The more features considered, the greater the demand on training data to produce accurate probabilities. However, [Back and Schwefel, 1993] proposes a method which uses probabilistic techniques to estimate probabilities for instances where the data is sparce.

Considering many features causes other difficulties. It can not be assumed that all features considered are independent of each other. Considering all interdepen-

dencies between features, leads to an extremely complex probabilistic model with a large number of parameters. [Gale *et al.*, 1995] propose Bayesian Discrimination Analysis to operate in a high dimension search space such as this. An alternative approach to the problem of interdependencies between features is proposed in [Bruce and Wiebe, 1994] and [Pedersen *et al.*, 1997]. Bruce and Wiebe use decomposable probabilistic models which do not consider all of the interdependencies between features. Using training data, Bruce and Wiebe identify which decomposable models are most beneficial for the disambiguation task.

An alternative statistic approach is cited in [Yarowsky, 1994]. Yarowsky uses statistical decision lists rather than adopting a Bayesian approach. Although many informative features are still identified, the classification of a word is based solely on the single most reliable piece of evidence in the context. Not attempting to combine evidence from different features, the problem of interdependencies between features is avoided.

In order to achieve accurate probabilities a large amount of training data is required and this is a major drawback for all approaches using probabilistic models. In most cases the required training data must be sense tagged and this substantially increases the required investment of human resources.

3.8 One Sense Per Discourse

This section considers one possible factor which may reduce the difficulty of solving the WSD problem. Following an evaluation of WSD algorithms [Gale *et al.*, 1992b], an investigation into a "One sense per discourse" hypothesis was performed. The following hypothesis was proposed: If a word appears in a discourse referring to a particular sense, it is unlikely that other senses of the same word will be referred to within the same discourse. Initial experimentation to test this hypothesis is reported in [Gale *et al.*, 1992c]. The experiment was conducted on a small scale, and considered nine ambiguous words. 54 pairs of sentences were identified. Both sentences in each pair contained the same ambiguous word and both were taken

from the same discourse. The results showed that 51 out of the 54 pairs referred to the same sense of the ambiguous word. Therefore, there existed a 94% probability that two instances of an ambiguous word would refer to the same sense in a given discourse.

The claims made by Gale et al are questioned in [Krovetz, 2000]. Krovetz notes that the accuracy of the "One sense per discourse" hypothesis depends both on the length of the discourse, and the level of similarity between the senses. Krovetz used SEMCOR and the DSO corpuses, which both tag words with WordNet senses, to test the hypothesis. Krovetz found that in 33% of the instances tested more than one sense of a word appeared in the same discourse. This is significantly higher than the results reported by Gale et al. The difference is accounted for by the fine level of sense distinctions made in the WordNet lexicon.

Krovetz's work shows that a strict "One sense per discourse" rule can not be applied to WSD. However, this does not imply that Gale et al's findings are not beneficial for WSD. As a result of the work of Gales et al, developers may decide that it is beneficial to consider all instances of a word within a discourse at the same time. The developer need not assign the same sense to all instances, but may use a weighting to prefer similar senses. It may also become beneficial to adopt different weightings for different words. As Krovetz mentions, the likelihood of more than one sense of a word appearing in the same discourse is dependent on the range of domains within which the senses can be used. This approach may enable instances where the ambiguity can be easily resolved to aid the disambiguation of other instances of the same word found in the same discourse.

3.9 Summary

This chapter has detailed many different sources of knowledge which can be used to help resolve word ambiguity. Various ways of using information from a dictionary and a thesaurus have been considered. Moreover, other contextual features such as morphology, collocations, part-of-speech, neighbouring words and syntac-

tic information have all been used by many of the systems considered. The recent availability of sense tagged data has also enabled stochastic information to be exploited.

The importance of combining different knowledge sources is noted in [McRoy, 1992], [Hwee Tou Ng and Hian Beng Lee, 1996] and most recently in [Wilks and Stevenson, 1999]. There exist many sources of information which are able to help disambiguation, but none are able to achieve high accuracy independently. Compared with other fields within NLP, the availability of many sources of information is an unusual feature for word sense disambiguation. The idea of employing weak knowledge sources for strong results must therefore be adopted. The challenge facing developers is to determine which knowledge sources should be adopted and also how these knowledge sources should be combined. The systems discussed have used a wide variety of techniques for combining knowledge sources. Harley and Glennon's system uses additive weights similar to those adopted in the evaluation of chess positions by computers. The combination of knowledge sources is optimised by a learning algorithm in the work of Wilks and Stevenson's [Wilks and Stevenson, 1998]. For the stochastic approaches combining the knowledge sources is not the problem, but handling the interdependencies between them. Wiebe and Bruce approach the problem using Decomposable Models which reduce these interdependencies. Yarowsky proposes Baysian Discrimination Analysis which is able to cope with a large number of interdependencies.

Finally, [Yarowsky, 1994] proposes a radical solution to the problem avoiding the need to combine knowledge sources. Yarowsky proposes that knowledge sources are useful in different instances. Therefore, the system must identify which knowledge source is most beneficial in a particular instance.

Chapter 4

Large Scale Knowledge Sources

The previous chapter examined in detail a range of knowledge sources which have been used by systems to aid the resolution of lexical ambiguity. This chapter follows on from chapter 3 by continuing to discuss the issue of knowledge sources. This chapter details the two knowledge sources used by the system being developed. Arguments for the choices made are given and compared with some of the different approaches discussed in chapter 3. Throughout this and subsequent chapters, this system will simply be referred to as DURHAM. DURHAM is the name given to this system for the SENSEVAL evaluation and is therefore an appropriate choice.

The two sources of information used by DURHAM to aid disambiguation are frequency and contextual information. The frequency knowledge source is commonly found in similar forms in other systems. However, the contextual information knowledge source is unique and is one of the main contributions made by this work. A third knowledge source which has been named **clue words** has also been developed. Clue words provide very reliable information, but difficulties exist with their ability to be applied on a large scale. Therefore, they do not feature in the knowledge sources considered for a large scale system which is the subject of this chapter. Chapter 7 discusses how the large scale system can be applied to a smaller domain. Therefore, clue words are considered in chapter 7.

The frequency and contextual information knowledge sources are combined to

produce a hybrid system. By doing so, both corpus based and sub-symbolic learning methods are exploited. The difficulty and nature of the task varies greatly for different words and is a reason why a hybrid approach is appropriate for the WSD task. For example, the number of different senses, the frequency distribution of those senses, the number of training examples available and the number of collocates which can help disambiguation all vary greatly for different words. Each of these factors affect the complexity of the task. As a result, the issue regarding the best method to combine individual knowledge sources is a difficult problem. The approach adopted by DURHAM and the reasons for the choices made are discussed in section 4.6. However, before the combining of knowledge sources can be discussed, both knowledge sources must be considered individually. Section 4.2 discusses the frequency information and sections 4.3, 4.4 and 4.5 discuss the contextual information and the method by which it is learnt. Before either of the knowledge sources are discussed an examination of the corpus used to train and test DURHAM is given.

4.1 Training and Test Data

Both of the knowledge sources used by DURHAM require manually sense tagged data to be trained and evaluated on. DURHAM is a large scale system. To fulfil the large scale criterion for success set out in section 2.7, DURHAM must be able to attempt to disambiguate all words in a lexicon. Therefore, the training and test data must also be large scale. Large scale refers to the number of different words which are sense tagged in the corpus. Large scale does not refer to the number of instances one particular word has been sense tagged.

There is only one corpus available to this project which fulfils these criteria. SEMCOR is a widely available manually sense tagged corpus which can be considered large scale. SEMCOR is a textual sense tagged corpus, so every open class word is assigned a WordNet sense tag. DURHAM uses the SEMCOR version which accompanies WordNet version 1.6. This version contains 359,732 words, of which

192,639 are open class words that have been assigned a WordNet sense. There are over 37,000 different senses of words identified in SEMCOR. A further section of SEMCOR is available, in which only the verbs are sense tagged. This section of SEMCOR is not used by DURHAM.

The utilised section of the SEMCOR corpus is split up into 186 files. There are approximately 100 sentences in each file. The files are grouped depending on the topic domain of the sentences contained within each file. The corpus is split into three sections:

- Training data
- Validation data[1]
- Blind test data

There exist 103 files used for training data which constitutes a section of SEMCOR known as *Brown1*. Out of the remaining 83 files, 30 are used as validation data and the remaining 53 files are blind test data. The files are split in order that there exists roughly equal proportions of each topic domain in the training data, validation data and blind test data. The role of each of these data sets is detailed in section 4.4.

4.2 The Frequency Knowledge Source

Dissimilar to some of the approaches discussed in section 3.7, DURHAM does not consider features as part of the frequency knowledge source. Some of the information which the stochastic features aim to encapsulate into a WSD system have been incorporated into different knowledge sources in DURHAM. This information includes collocations and semantic tags. Other features such as the POS of neighbouring words have not been incorporated into DURHAM. The POS of neighbouring words is not a primary source of evidence to aid WSD, but is employed by

[1]This could also be considered semi-blind test data

some systems as a backup mechanism when no other means are available [Yarowsky, 1996]. By incorporating the POS of neighbouring words, the amount of training data required is greatly increased. The sense tagged training data available for a textual corpus gives limited examples for each word. A textual corpus is required as training data for DURHAM to keep in line with the large scale criterion already set out. As a result, the increase in accuracy achieved by considering the POS of neighbouring words is lost by the resulting less accurate stochastic information.

As a consequence of eliminating the use of features, the frequency information is not based on Bayes Theorem and consideration of the interdependencies between features is not necessary. Instead, the frequency information is used simply to measure the likelihood of each possible sense appearing in the text. Counting the number of occurrences for each sense in the sense tagged corpus is the basic method to calculate the statistical information. The number of occurrences for all possible senses is assigned at the beginning of the program. This ensures that during training, the frequency information can be obtained without computational expense. This method is slightly modified for the SENSEVAL task. These modifications are discussed in chapter 7. For the large scale system, the frequency information is obtained solely from the section of SEMCOR assigned as training data. This policy has been strictly enforced. For example, WordNet assigns a number for each sense for every word in the lexicon. The lower the number assigned, the more frequent the sense. Although this source of knowledge is very valuable for a WSD system, it can not be employed in DURHAM. The reason for this is that the entire SEMCOR corpus will have been used to determine the number for each sense. Therefore, the WordNet numbers are determined using the blind test data which is not considered acceptable. The formula used to calculate the frequency information is given in section 5.1.

Although frequency information is a straightforward knowledge source, it is still very useful. Most lexicons contain senses which are obscure and infrequently used. The frequency information enables the more commonly used senses to be favoured. For example, in SEMCOR there are 2,345 noun instances of the word *group*. Out of these instances, 2,333 refer to the *members considered as a unit* sense. The

chemical sense is referred to in nine instances and the *blood* sense is referred to in three instances. Therefore, solely using the frequency information alone 99.49% accuracy can be achieved for *group* on this data set. The accuracy achieved by solely using the frequency information is known as the **frequency baseline**.

The frequency information also helps to direct the training of the contextual information. This contributes to enabling the overall system to achieve accuracy above the frequency baseline.

4.3 Contextual Information

This section discusses the knowledge source referred to in this work as **contextual information**. The format of the knowledge source is novel and is one of the major contributions of this work. Contextual information is based on the WordNet hierarchy and uses a sub-symbolic learning mechanism. Therefore, like the frequency information, the contextual information requires training data. A justification for the design of the contextual information will be given. This is followed by a description of the algorithm used to learn contextual information.

4.3.1 Aims for Contextual Information

A fundamental component for the development of contextual information was the recognition of WordNet as a valuable resource for WSD. WordNet is discussed in detail in section 3.5. However, the advantages and disadvantages of using WordNet to aid WSD can be summarized as follows:

√ WordNet provides a fine grained sense inventory.

√ Synsets and the low level WordNet hierarchy are beneficial in reducing the required training data.

√ SEMCOR is the only large scale, widely available sense tagged corpus. SEMCOR is sense tagged with WordNet senses.

X Problems exist with the high level WordNet structure for WSD. This is highlighted by the tennis problem.

X In general words of different POS are unconnected in the WordNet hierarchy.

X The manual sense tagging accuracy of SEMCOR is not as high as would be hoped. This is measured in terms of ITA.

To benefit from the useful characteristics of WordNet and to try and overcome the problems also associated with WordNet is the aim of the contextual information knowledge source. To help overcome the identified problems, SEMCOR is used as a training resource. However, problems have also been identified with using SEMCOR for WSD, and these are also considered. The structure of the contextual information knowledge source will now be explained. The justification for the choices made relate back to these aims.

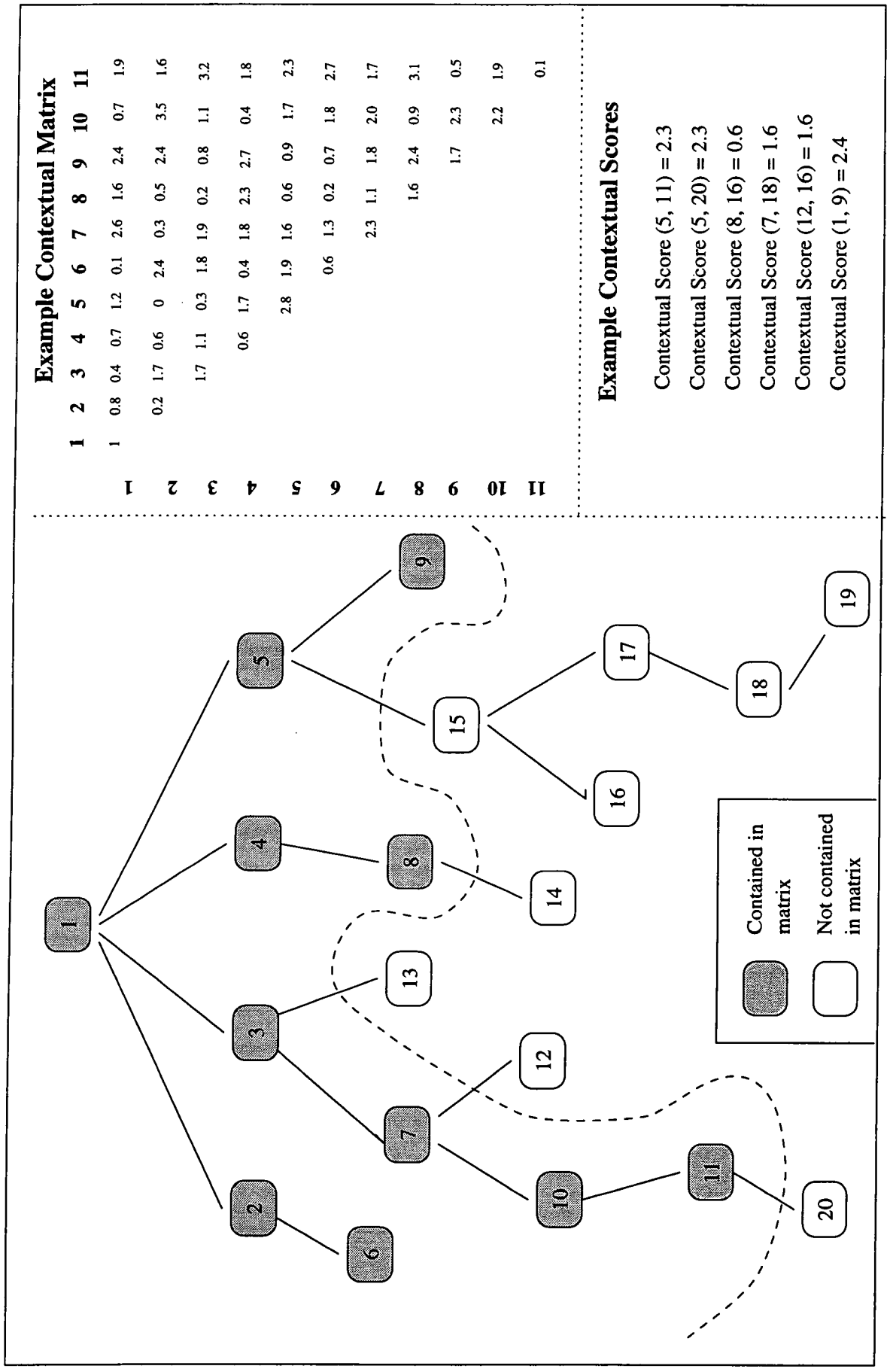
4.3.2 The Contextual Score

Contextual information is concerned with learning contextual scores between nodes in the WordNet hierarchy. Figure 4.1 is used to show how a contextual score is calculated. Just under 2000 of the high level concepts in WordNet are represented in a **Contextual Matrix**, and the contextual matrix stores scores between all of these nodes. The reason for this number, and the method by which these nodes are selected is detailed later in this section. The section also compares this method with other similar approaches.

To calculate the contextual score between any two nodes in WordNet, the presence of both nodes in the matrix is checked. If either of the nodes do not appear their hypernyms are moved up until a node is found which is in the matrix. The score between the two appropriate nodes in the matrix can then be obtained.

Nodes from all four of the WordNet hierarchies - nouns, verbs, adjectives and adverbs are included in the contextual matrix. By including all POS in the contextual matrix, the four hierarchies become much more connected. This overcomes

Figure 4.1: Diagram showing how to calculate the contextual score between two nodes in WordNet



one of the problems identified with using WordNet, the lack of connection between POS hierarchies. Greater connection enables the training process to learn contextual information between words of different POS. Possessing scores between senses of different POS enables all open class words found in the sentence to provide contextual information and assist in resolving the ambiguity of a word. It also ensures that all ambiguous words regardless of their POS can be disambiguated. The later benefit is important in order to fulfil the large scale criterion set out for DURHAM. DURHAM will only be considered large scale if words from all POS can be disambiguated.

It is the lack of connection between the different POS hierarchies which prevents the system reported in [Agirre and Rigau, 1995] disambiguating all ambiguous words. Agirre uses a semantic distance measure based on the WordNet hierarchy. Agirre is only able to disambiguate nouns. Moreover, Agirre is only able to use nouns to provide contextual information.

The low level concepts in WordNet are not included in the contextual matrix. The reason for this is that it is not considered beneficial to try to learn scores between highly related concepts. As shown in [Leacock and Chodorow, 1998], the low level WordNet hierarchy and the synset structure can be used to reduce the requirement on training data. Therefore, concepts represented by the same node in the contextual matrix will all contribute to learning the contextual scores for that node. As contextual scores generally operate between concepts above the word level, more general information is taken from each training instance.

The low level WordNet hierarchy is considered more beneficial for WSD than the classification of concepts higher up the hierarchy. For WSD, problems exist with the high level WordNet hierarchy. This is highlighted by the tennis problem. All the concepts associated with tennis are distributed widely over the hierarchy. Therefore, these concepts are not considered similar through the classification adopted by WordNet. However, these concepts are likely to appear together in the same sentence and provide useful contextual clues to aid WSD.

[Karov and Edelman, 1996] consider the reliance on the structure of a lexical

hierarchy the major drawback for a semantic distance approach, such as those considered in section 3.6.2. Karov and Edelman highlight this drawback using *hospital* and *doctor* as their example. The high level WordNet structure determines that these two words have a high semantic distance. *Hospital* is classified as a type of building and *doctor* is classified as a type of professional. However, if found in the same sentence, *hospital* is an excellent contextual clue to identify the correct sense of *doctor*. This example highlights another problem with the high level WordNet hierarchy. All senses of doctor are classified as people. As a result, all senses are found in a similar section of the WordNet hierarchy. This makes it increasingly difficult to use the hierarchy to distinguish between the senses. An attempt to develop a more beneficial hierarchy specifically for WSD was undertaken by the CoreLex project [Buitelaar, 1998]. Using the WordNet lexicon, 126 semantic tags were identified which aimed to distinguish between the homonymous senses of ambiguous words.

The purpose of the contextual score is to determine a more appropriate classification specifically for WSD. This classification is learnt using SEMCOR. As a result, a contextual score is different from a semantic distance. A contextual score aims to represent the likelihood of two concepts appearing in the same sentence. Semantic distance represents the extent to which two concepts are semantically similar based on a lexical hierarchy.

Two difficulties arise with developing a contextual score in this manner. Firstly, the determination of the number of nodes which should be included in the contextual matrix. Secondly the identification concerning which nodes should be included in the contextual matrix.

Selecting Nodes for the Contextual Matrix

1973 nodes are included in the contextual matrix so that in most cases, all senses of a word are represented by a different node in the contextual matrix (A justification for the number of nodes in the contextual matrix is given later). If sufficient nodes were included so that all senses were represented by different nodes, almost all nodes

in the WordNet hierarchy would need to be included. Such a large contextual matrix would place a much greater demand on training data. As a result, the contextual information is unable to make some fine grained sense distinctions. This is not a major drawback for the contextual information knowledge source. The purpose of this knowledge source is to assist in distinguishing between more coarse grained sense distinctions. In most cases, coarse grained sense distinctions are more important than fine grained distinctions. Many NLP tasks do not require the ambiguity of a word to be resolved to such a fine level. Moreover, there exists a low ITA between sense taggers, and this is one of the problems identified with using SEMCOR. A great deal of the disagreement between sense taggers will be between finely grained senses. If two finely grained senses are represented by the same node in the contextual matrix it is less important if these senses are inconsistently sense tagged in SEMCOR. Therefore, the credibility for using SEMCOR as training data, despite its low ITA, is increased.

To determine which nodes should be included in the contextual matrix a frequency measure is adopted. This is similar to the approach adopted in [Resnik, 1995a]. The depth of the node in the WordNet hierarchy does not reflect how specific the concept is which the node represents. This is because the WordNet hierarchy is unevenly distributed. Figure 4.1 shows how the cut off point for a concept's inclusion in the contextual matrix can occur at varying depths. The selection process is carried out using the section of SEMCOR assigned as training data. For each occurrence of a WordNet node in SEMCOR, that particular node, and all the hypernyms of that node are incremented by 1. This produces non increasing frequency counts as the WordNet hierarchy is moved down. For nouns and verbs, nodes are included in the contextual matrix only if their total frequency count is over twenty. However, the frequency cut off point for adjectives and adverbs was ten. The reason for this is that adjectives and adverbs occur less frequently in the training text and there is less hierarchical structure for them in WordNet. If an adjective or adverb synset occurs more than ten times in SEMCOR then that synset and the antonym synset are included in the hierarchy. Infrequently occurring adjective/adverb synsets are all represented in the contextual matrix by an

adjective/adverb root node.

It should be noted that it is not the aim of this work to find an optimum number of nodes for the contextual matrix, or an optimum method for selecting those nodes. The choices made are based on subjective decisions. This work only claims that the choices made do work. It does not claim that other choices will work better or worse. Also this work uses all the words in the sentence as a context window. This is also a subjective choice and does not claim that the choice is better than either a larger or smaller context window.

4.4 Learning Contextual Scores

Now that the concepts of a contextual score and a contextual matrix have been detailed, the mechanism through which contextual scores are learnt is considered. Before learning commences, all scores in the matrix are set equal to each other. This is carried out in order that no pre-conceptions for the contextual scores are made before training commences. An alternative approach may be to use a semantic distance measure between concepts as a starting point for a contextual score. The benefits of such an approach are considered. If a semantic distance measure does provide useful contextual information, then the starting point for learning is higher than if all nodes are set equal to each other. This could be beneficial particularly with limited training data for each ambiguous word. The learning process could then be concerned with optimising semantic distances. A high semantic distance represents a poor contextual clue but a high contextual score represents a good contextual clue. Therefore, the semantic distance measure would need to be inverted.

Two reasons exist why semantic distances are not used as a starting point in this work. Firstly, the semantic distance measure is meaningless between concepts with different POS. As a result, it would be difficult to derive an appropriate starting point for contextual scores between senses of different POS. Potentially this could lead to the contextual information from different POS not contributing equally to

aid the resolution of an ambiguous word. Secondly, the process adopted to learn contextual scores is sub-symbolic. As with most sub-symbolic learning mechanisms, the internal mechanism for determining the solution can not be understood by the developer. This is referred to as a black box architecture. Initiating the learning process with semantic distance scores may not be appropriate for this architecture. As a result, the learning process may be forced to a local maxima from which it can not escape. Setting all scores equal to each other is a more conventional approach to initiating weights for sub-symbolic learning.

The first stage of the training process involves taking a single sentence from the training data. The sentence is disambiguated using the algorithm discussed in chapter 5. The algorithm returns a sentence with all open class words assigned with a WordNet sense. The sense tagged sentence is compared with the manual sense tags assigned for each word. The learning process identifies which words were incorrectly sense tagged. More importantly, it also identifies which words in the sentence used as context provided evidence for any incorrect classification. This is carried out by comparing the contextual score between each context word and the correct sense, with each context word and the chosen sense. This information is used to calculate the amount contextual scores should be changed. The algorithm for changing contextual scores is detailed in section 4.5.

Nodes in the contextual matrix are typically above the word level. This is likely to decrease the required amount of training data. However, further steps are taken to extract as much information as possible from each training sentence. In order to achieve this, further generalisations are carried out also. Not only are contextual scores between nodes which represent words in the training text changed, but so are their hypernyms and hyponyms.

Once all training sentences have been processed, the new contextual matrix is tested on the validation data. If an improvement in accuracy is observed then another iteration of the training phase will be initiated. This is repeated until there are no more improvements on the validation data. Finally, DURHAM is tested on blind test data which has until that point been unseen by the system. The learning

process is summarized in figure 4.2.

4.4.1 Learning Contextual Scores Example

The learning algorithm is now further explained with the aid of an example. The example aims to highlight some of the benefits of adopting this approach. The example considers the ambiguous word *board*. This word is also used as an example in [Voorhees, 1998] and [Voorhees, 1993]. Voorhees adopts a semantic distance approach based on a *hood* which is a section of the WordNet hierarchy that subsumes one possible sense for an ambiguous word. Disambiguation then proceeds based on the following principle:

> *"We use the hoods of the synsets containing an ambiguous word w to define the categories that represent the different senses of w. Another word that occurs in a text with w and is a member of a synset in the hood of one of the senses of w is evidence for that sense of w."*
>
> [Voorhees, 1998] page 291

The approach adopted by Voorhees will be used in the example for comparative purposes.

The example assumes a training sentence such as *"I hit the board with my hammer"*, where *board* is manually sense tagged to the *Board(plank)* sense. The first stage of the learning algorithm is to use the disambiguation algorithm to automatically sense tag the training sentence. For the purposes of the example let us also assume that the disambiguation algorithm incorrectly assigns the *Circuit board* sense. Figure 4.3 shows the senses of *board* taken from WordNet that are considered in this example.

The approach adopted by Voorhees does not use training data, and therefore must rely on the original WordNet structure. This is shown in figure 4.3. Using the Voorhees method, *Device* will be the hood for the *Circuit board* and *Control*

Figure 4.2: Flow graph to illustrate the contextual score learning mechanism

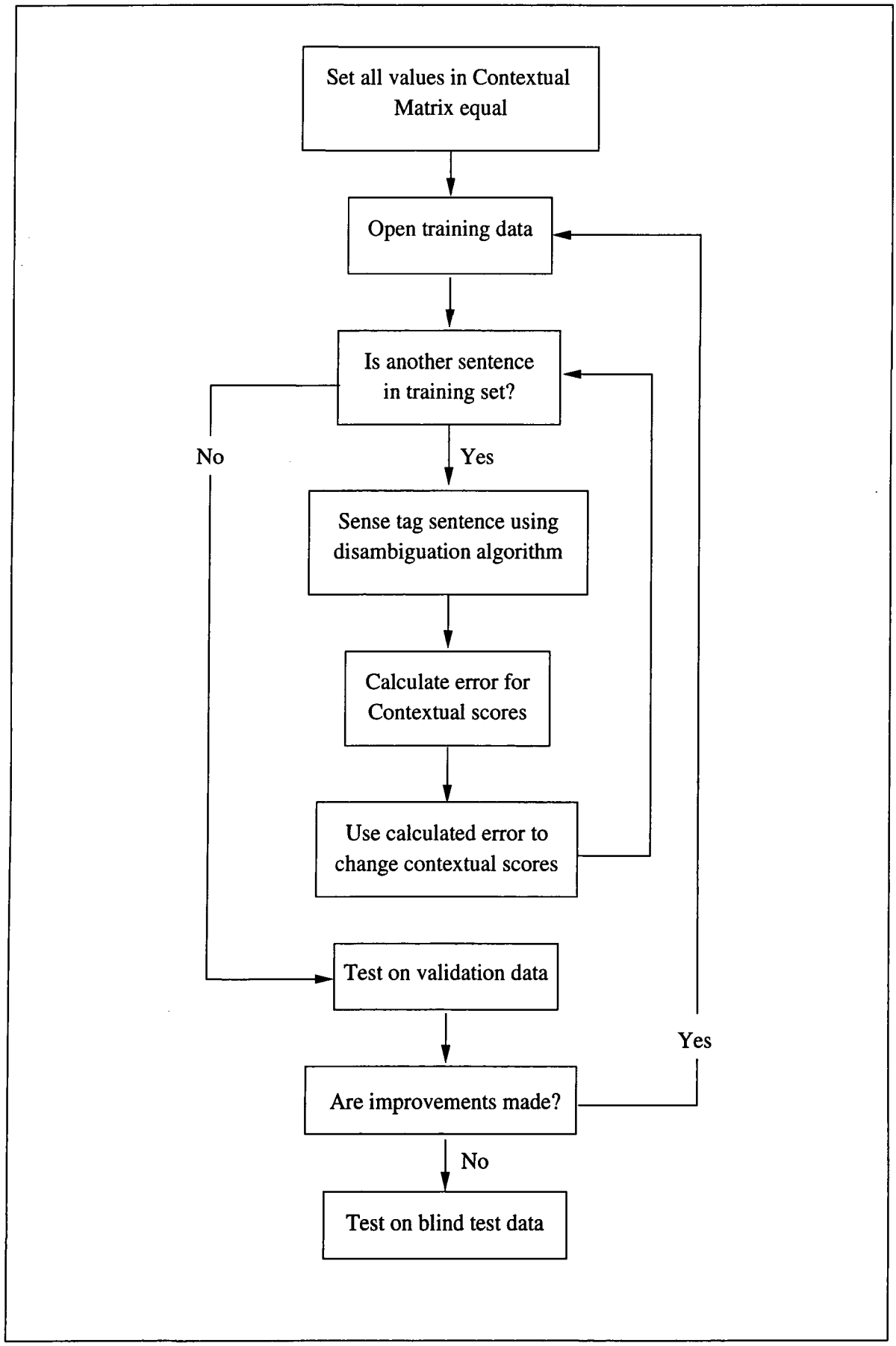
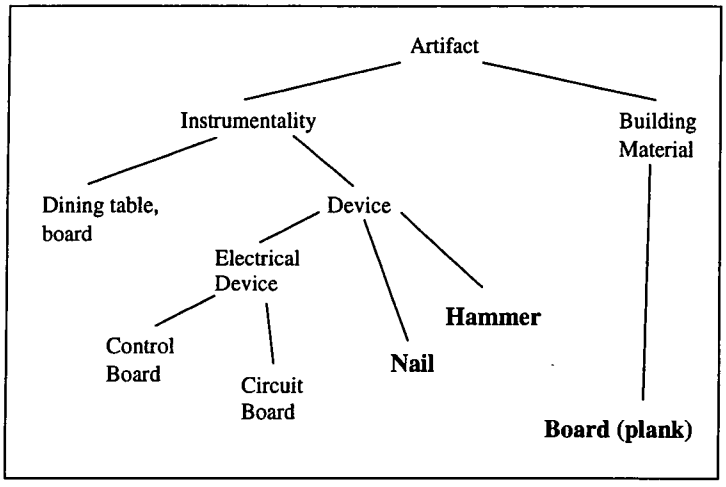


Figure 4.3: Diagram showing the original WordNet structure before learning.

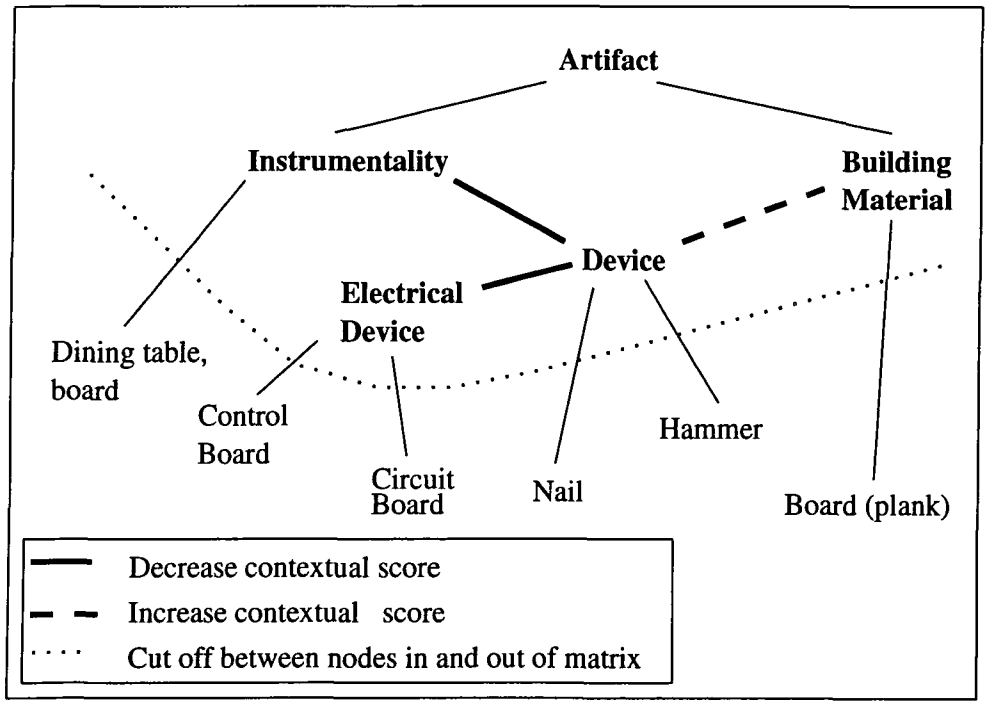


board senses. As *hammer* and *nail* are both members of this hood they will incorrectly provide evidence for these two senses. This highlights the problem with the semantic distance approach.

Unlike a semantic distance approach, DURHAM is able to use the training data to improve the disambiguation accuracy. In this example the words *hit* and *hammer* are used as context to help disambiguate *board.* For simplicity, the example will only consider the context word *hammer*, but the same method can be repeated for *hit. Hammer* is represented by *Device* in the contextual matrix. The correct sense of *board* is represented by *Building Material* and the incorrectly chosen sense is represented by *Electrical Device* in the contextual matrix. The training process will then increase the contextual score between *Device* and *Building Material* and decrease the score between *Electrical Device* and *Device.* Thus making *hammer* a better contextual clue for *Board (plank)* and a worse contextual clue for *Circuit Board.* A high contextual score represents a good contextual clue. These changes in contextual scores are shown in figure 4.4.

This example highlights that generalisations are made during training by only including the higher level WordNet nodes in in the contextual matrix. By increasing the contextual score between *Device* and *Building Material*, *Nail* will automatically receive a higher contextual score with *Board (plank).* Moreover, decreasing the contextual score between *Device* and *Electrical Device* automatically decreases

Figure 4.4: Diagram showing how contextual scores change if *hammer* and the *board (plank)* sense of board appear in a training sentence.



the scores between *Hammer* and *Control Board.* This shows one way in which generalisations are made to extract extra information from each piece of training data.

The further generalisations within the contextual matrix are also shown in Figure 4.4. The contextual score between nodes *Instrumentality* and *Device* is reduced as *Instrumentality* is a hypernym of *Electrical Device.* The *Electrical Device* node in the matrix represents the incorrectly chosen sense of *board.* In this example, the generalisation mechanism enables the *Dining table* sense of *board* to reduce its contextual score with *Hammer* and *Nail.*

If the *Dining table* sense of *board* had been chosen by the disambiguation mechanism, the contextual score of *Electrical Device* would still have been reduced. This follows because *Electrical Device* is a hyponym of *Instrumentality.* Contextual scores involving *Artifact* are not moved because it is contained in the hierarchy of both the chosen and correct sense of *board.*

The net result of the training sentence is that *nail* and *hammer* become better contextual clues for the *Board (plank)* sense.

4.5 Changing Contextual Scores

The features of the learning mechanism have now been examined. The learning mechanism has been shown to make generalisations to reduce the reliance on a large quantity of training data. It has also been shown to be less dependent on a semantic hierarchy as some semantic distance approaches. This section continues to examine the learning mechanism by detailing the method adopted to change contextual scores.

The disambiguation mechanism discussed in chapter 5 provides the information for determining how the contextual scores should be changed. Two pieces of evidence are used to determine which contextual scores should be changed and to what degree these should be altered. The first piece of evidence identifies whether the ambiguous word has been assigned the same sense as was manually identified. The second piece of evidence identifies the amount the words serving as context contributed towards any misclassification of a sense.

To calculate the extent to which a score should be changed between nodes, an error function is determined. The error represents the difference between the contextual score of the correct sense and the context sense, and the contextual score of the chosen sense and the context sense. If the chosen sense is correct then the error is equal to zero.

$$Error = CS(correct, context) - CS(chosen, context) \tag{4.1}$$

Where CS is the contextual score. The error is then used to calculate the amount each contextual score should change. The change is calculated as the sigmoid function of the error.

$$Change = \frac{1}{1 + e^{-L*error}} \tag{4.2}$$

The sigmoid function is chosen because of the similarities between this type of learning and the error distribution mechanism of Back Propagation (the standard learning algorithm in Neural Networks). Error back propagation uses the sigmoid function. It enables contextual scores between nodes which have caused high error to be altered more than those with small error. This is done without allowing any scores to change substantially as this could cause oscillation. L is the learning rate which is reduced throughout the learning phase, so changes to scores become less. This is similar to Simulated Annealing also used in Neural Networks.

The learning algorithm adopts the same method to take advantage of senses which have been disambiguated correctly. In such cases the correct and chosen sense are the same leading to the error being equal to zero. Using equation 4.2, the change will be 0.5. This slightly reinforces the contextual score between the correct sense and contextual word. By doing so it helps to ensure that DURHAM will continue to correctly classify the senses which it was able to classify before training.

The calculated change is then added to the contextual score between the correct and context sense, enabling the context sense to provide stronger evidence in the future. The change is also subtracted from the contextual score between the chosen and the co

ntext sense, making the evidence for selecting the incorrectly chosen sense weaker. The change score is also used as the basis by which the hypernym's and hyponym's contextual score will also be altered. Using the training data to change the scores between hypernyms by making further generalisations is not as reliable as changing the initial nodes. As a result, the contextual scores are moved by less. The contextual score of a hypernym or hyponym is changed by half the amount that the initial node was changed to compensate for this. A grandparent/grandchild of

a node is changed one quarter of the amount, and so on.

4.6 Combining Knowledge Sources

One problem facing all developers who choose a hybrid approach to WSD is the combining of the knowledge sources they incorporate. The amount each knowledge source is able to help disambiguation varies for each particular word. By combining these knowledge sources the aim is to take the useful information each is able to offer, and restrict the confusion in cases where they are unable to help. Various approaches to this problem were considered in chapter 3.

DURHAM chooses additive weights to combine knowledge sources as used in [Harley and Glennon, 1997]. Adding scores from each knowledge source is more appropriate than multiplying them in this instance. Multiplying scores is beneficial only if the scores represent the probability of a particular outcome. However, the contextual score is not a probabilistic measure. Also, there are many senses of words which appear in the blind test set which do not appear in the training set. For these senses the frequency score will equal zero. If the scores were multiplied, the overall score would equal zero regardless of the contextual information's score. Adding scores also enables the possibility for the contextual information to assign a negative score. Moreover, adding scores seems more appropriate than Yarowsky's method that chooses the single best knowledge source for each instance. By assigning a very high score to one knowledge source, that knowledge source can have the overriding outcome on the choice of a sense. However, adopting the additive framework also allows a combination of evidence in cases which are less clear cut. This is not possible using Yarowsky's method.

4.6.1 The Roles of each Knowledge Source

For the DURHAM large scale system, both the frequency and contextual information have been trained from the same training data. This section discusses why

the two knowledge sources can not be combined into one knowledge source. In addition, it discusses the steps taken to ensure that the knowledge sources are extracting different information from the training data. The central issue to both of these questions is an understanding of the specific role each knowledge source aims to achieve.

The frequency information provides fine grained evidence as it operates at the sense level. For the frequency information to be beneficial in resolving the ambiguity of a word, there must exist a large number of training instances containing that word. Even if there are a large number of training instances, the frequency information will only be beneficial if the distribution of senses is skewed. The contextual information complements the frequency information by operating above the word level. Therefore, it provides more coarse grained evidence. If more than one sense is represented by the same node in the contextual matrix, the contextual information has no way to determine between them. A choice between the possible senses must be taken from the frequency information. By operating above the word level, the contextual information is less reliant on the coverage of the training data. The aim of contextual information is to find clues in the surrounding context to resolve the ambiguity. As a result, it is less dependent on the frequency distribution of a word's senses. Therefore, the two knowledge sources aim to aid the ambiguity resolution of different types of words. This is the reason why separate knowledge sources are required.

The frequency information is calculated before the contextual information so that it can be used during the training of the contextual information. This helps to prevent the contextual information simply learning to choose the most frequent sense of each word. In the early stages of the contextual information training process, the system is likely to choose the most frequent sense. Misclassifications will take place in instances which do not refer to the most frequent sense. These misclassifications aid the contextual information in learning contextual evidence for the less frequent senses. Identifying contextual evidence for all senses, not only the most frequent, helps to ensure that the contextual information fulfils its specific role and complements the frequency information.

The use of frequency information as a knowledge source has for a long time been inhibited by the influence of Chomsky's work [Chomsky, 1965]. He proposes a distinction between understanding the linguistic content of the problem and generating good performance. Relating his claims to this work, the frequency information goes some way to improving performance. However, he claims that frequency information does not aid the understanding towards the reason why a particular sense is chosen. A disambiguation system which aims to achieve 100% accuracy will not be able to use frequency information as a knowledge source. Frequency information enables guesses to become more educated. Seldom can frequency information always supply evidence for the correct sense.

For this work, there are several reasons why Chomsky's claims are not accepted. Firstly, the frequency information helps direct the learning of the contextual information and therefore aids the process of learning linguistic content. Secondly, a measure of how well the linguistic problems have been learnt can still be determined. This is done by examining the performance of the system with respect to the frequency baseline. The frequency baseline is the accuracy achieved by the system if the most frequent sense is always chosen. As developments are made to improve the contextual information, the extent to which the frequency information contributes towards accuracy is reduced. For the foreseeable future, setting a goal of achieving 100% accuracy is unrealistic. Finally, from a NLP perspective, generating good performance is the goal of the WSD system. The goal is not to develop an understanding of linguistic content.

4.7 Summary

This chapter has set out the three subsections of SEMCOR which are used for training and evaluating DURHAM. These three data sets are training data, validation data and blind test data. The chapter then proceeded to examine the two knowledge sources employed for evaluation on SEMCOR. The frequency knowledge source is straight forward and was only calculated from the training data.

The contextual information knowledge source is novel. The features of contextual information were discussed and the mechanism by which it is learnt was detailed. Finally, the method by which the two knowledge sources are combined was examined.

Chapter 5

The Disambiguation Algorithm

The characteristics of the knowledge sources used by DURHAM for evaluation on SEMCOR were detailed in the previous chapter. This chapter details the formula used to calculate the scores for the frequency and contextual information. The formula for combining the individual scores from each knowledge source is then detailed. This formula is used to assign each possible sense in the sentence with a score which represents the likelihood of that sense being chosen. These scores are then used to determine which sense will be selected for each ambiguous word.

The chapter then moves forward to detail the mechanism by which the scores are used to select senses for each word. This is not a trivial task. There are two problems which need to be considered. Firstly, the problem outlined by Wilks, detailed in section 2.3.1. This highlights the number of sense combinations that a sentence may possess. Therefore, the computational expense of the mechanism must be considered. A system which considers all possible sentence combinations will be too computationally expensive, particularly for long sentences. The second problem is related and is concerned with the circular nature of the disambiguation task. The contextual information for any word in the sentence is dependent on the sense choices made for other ambiguous words in the sentence. Therefore, for all words it is difficult to select the correct sense until all other words in the sentence have been disambiguated.

DURHAM adopts a novel approach to overcome these difficulties. After the disambiguation mechanism has been detailed, it will be compared with methods adopted by other systems to overcome the same difficulties.

The definitions of the mathematical notation used throughout this chapter is now given:

w_i	refers to the i_{th} word in the sentence.
w_{ij}	refers to the j_{th} sense of the i_{th} word in the sentence.
$Freq(w_{ij})$	is the frequency of sense j of word i in the training data.
$CS(w_{ij}, w_{kl})$	refers to the contextual score between w_{ij} and w_{kl}.
$CIS(w_{ij})$	The contextual information score assigned to sense w_{ij} by combining the scores from all the context words.

5.1 Calculating Scores

This section details the mathematical formulae used for calculating the scores for the frequency and contextual information.

The frequency score is the probability of a particular sense occurring given that one of the possible senses of the word has occured. The formula used is as follows:

$$Frequency_{ij} = \frac{Freq(w_{ij})}{\sum_{k=1}^{m} Freq(w_{ik})} \tag{5.1}$$

where m is the number of possible senses for word i. It is possible for a word to be found in either the validation data or blind test data, which has not been contained in the training data. In these instances the nominator and denominator in equation 5.1 are both equal to zero. This produces an undefined frequency score using equation 5.1. In these instances the $Frequency_{ij}$ is set to zero.

The calculation of the contextual score is more complex. Unlike the frequency information, the scores for each sense are dependent on the sense choices made for other words in the context window. The context window is one sentence, but this choice is not claimed to be optimal. The method by which the contextual score is

calculated is the first step towards overcoming the problem of knowing where to start disambiguating.

To calculate a score for a particular sense, no sense choices are assumed for contextual words. Instead, a maximum and minimum score for each sense is calculated based on the sense choices made for the contextual words. The process by which the maximum and minimum contextual scores for a sense w_{ij} are calculated is now detailed.

For each word w_k in the context of the sentence, the senses of that word which possess the greatest and smallest contextual score with w_{ij} are initially identified. The scores between these senses and w_{ij} are named $MaxContext$ and $MinContext$.

$$MaxContext(w_{ij}, w_k) = \max\{CS(w_{ij}, w_{k1}), ..., CS(w_{ij}, w_{km})\} \quad (5.2)$$

$$MinContext(w_{ij}, w_k) = \min\{CS(w_{ij}, w_{k1}), ..., CS(w_{ij}, w_{km})\} \quad (5.3)$$

For contextual words which are not ambiguous, the $MaxContext$ will equal the $MinContext$. The $MaxContext$ and $MinContext$ scores for all the contextual words in the sentence are added together to produce a maximum and minimum contextual information score for sense w_{ij}. This is shown in equations 5.4 and 5.5. These scores represent a measure of the strength of the evidence which all the combined contextual words assign to a particular sense.

$$Max\_CIS(w_{ij}) = \frac{\sum_{k=1k\neq i}^{n} MaxContext(w_{ij}, w_k)}{n} \quad (5.4)$$

The minimum contextual information score is calculated in a similar way.

$$Min\_CIS(w_{ij}) = \frac{\sum_{k=1k\neq i}^{n} MinContext(w_{ij}, w_k)}{n} \quad (5.5)$$

In equations 5.4 and 5.5, n is the number of open class words in the sentence.

Normalising the contextual information score by the number of words in the sentence is important. A long sentence will contain more contextual words than a short sentence. Normalization ensures that the value of the contextual information scores is independent of the sentence length. This does not effect the rank order of the senses with respect to the contextual information. However, normalization facilitates the combining of the contextual information score with the frequency score. The amount the contextual information score contributes to the overall score will be the same regardless of the sentence length.

5.2 Combining Scores

In the previous section, the process by which the frequency score and the maximum and minimum contextual information scores are calculated was detailed. This section continues on the same theme by examining the formula used to combine these scores. The formula uses additive weights to combine the knowledge sources. The weights represent the extent to which each knowledge source contributes towards the overall score.

It is not considered beneficial to multiply the scores from each knowledge source. Multiplicative weights are beneficial in probabilistic systems where each knowledge source assigns a probability for each sense. Although the frequency information score assigns a probabilistic measure for each sense, the contextual information score is not a probabilistic measure. Therefore, the benefit of multiplicative weights is lost. In addition, employing additive weights enables the contextual information to assign negative scores.

The previous section detailed how the contextual information assigns two scores for each sense, a maximum contextual information score and a minimum contextual information score. As a consequence, a maximum and minimum score is also produced for the overall score.

The formulae for calculating the overall maximum and minimum scores for sense w_{ij} are given in 5.6 and 5.7.

$$MaxScore_{ij} = A * Frequency_{ij} + B * Max\_CIS(w_{ij}) \quad (5.6)$$

$$MinScore_{ij} = A * Frequency_{ij} + B * Min\_CIS(w_{ij}) \quad (5.7)$$

A and B are constants which represent the weights which the frequency and contextual information contribute to the overall score. A possible area which could be examined in future work may include investigating any benefits associated with not using constants for these weights. This is because the amount each knowledge source is able to contribute towards each sentence is variable with each sentence. If this can be measured then the contribution of that knowledge source can be varied accordingly using the weights. For example, the frequency information provides much more reliable evidence for senses of a word if there have been many training instances. Therefore, future work could examine defining A as a variable, which is dependent on the number of training instances found for a particular word.

Using variable weights was not a direction of research developed in DURHAM because both knowledge sources are trained on the same data. The contextual information will also assist in providing better evidence if many training examples exist. If one knowledge source has strong evidence for a particular sense, then that knowledge source can assign a large score to the particular sense. This would be sufficient to ensure that the particular sense is chosen without the need to change constants A and B.

Finding the optimum values for the constants A and B is also beyond the scope of this thesis. The optimum values for A and B during the training phase may not be the same as the optimum values for testing. Therefore, DURHAM would need to be trained many times using different values for A and B in each instance. Each of these instances would then need to be tested again using different values for A and B. Section 6.4 examines the affect on the accuracy of varying the values of A and B. This analysis is beneficial in examining the contribution each knowledge source makes to the overall accuracy of DURHAM. However, the analysis also suggests

that the optimum values for A and B may not generate an increase in accuracy which is statistically significant.

5.3 Eliminating Senses

The previous sections have examined the process by which a maximum and minimum score is generated for each possible sense in a sentence. This section progresses on from sections 5.1 and 5.2 and details the mechanism by which the maximum and minimum scores are used to determine the sense which should be chosen for each word.

This is a difficult problem due to the interdependencies between the sense choices of all the ambiguous words in the sentence. Eliminating a sense of one context word, may greatly reduce the score for a sense of another word. The problem can be considered at two levels, the word level and the sentence level.

At the word level, the difference between the maximum and minimum scores for all the possible senses represents the level of uncertainty remaining in the system. As senses of context words are removed, the difference between the maximum and minimum scores for each sense is reduced. This removal of some of the uncertainty makes it easier for the word to eliminate one or several of its possible senses. As a result, more of the uncertainty for the context senses is eliminated so more of their ambiguity can be resolved. This unravelling process proceeds until all the ambiguity is resolved. Each step reduces the uncertainty in the system making subsequent steps more reliable. Once the sense choice has been made for all ambiguous words in the sentence, the maximum and minimum scores will be the same as no more uncertainty remains in the system.

At the sentence level, the purpose of this mechanism is to try to find the combination of senses for each ambiguous word which generates the highest total score for the sentence. The score for the best sense combination will be referred to as the **best sentence score.** As some sentences may have many million different sense combinations, identifying the best sentence score is not a trivial task. A mechanism

which considers all possible sentence combinations to find the best sentence score is not practically viable. For long sentences, the computational expense of such a mechanism is far too great.

Moreover, a simple mechanism which chooses the sense for each word with the highest max score or a similar heuristic would also be unsatisfactory. It is important to consider which sense for each of the context words is chosen, as this influences the information the context words are able to provide. A sense (w_{ij}) could have a high max score because it possesses a high contextual score with a particular context sense (w_{kl}). If the context sense w_{kl} is not chosen for w_k, then the max score for w_{ij} will be reduced. The effect that using the correct sense for the context words makes to the overall accuracy of DURHAM, is considered in section 6.6.

The mechanism adopted by DURHAM to select a sense for each ambiguous word is novel. The iterative process is directed by considering scores at the word level, as this is substantially less computationally expensive than considering sentence scores. This seems particularly appropriate for the knowledge sources used by DURHAM as the frequency information is not dependent on the sense choices made by the context words and therefore operates solely at the word level. However, the aim of each iteration is to reduce the possible number of sense combinations in the sentence without making unjustified assumptions about the correct senses of context words. By so doing the senses eliminated in each iteration are those which are least likely to contribute to the best sentence score. Therefore, the aim is to identify a sentence combination with a score as close to the best sentence score as is possible, without considering all of the possible combinations. In instances where the best sense combination is identified, the ambiguity resolution may still not be correct. However, the cause of a correct sense being eliminated is an inappropriate score assigned to the sense by the knowledge sources, not the elimination algorithm itself.

This mechanism adopted ensures that only the senses which have not been eliminated are used as contextual information for other ambiguous words. After each iteration, the maximum and minimum scores for the remaining senses need to

be recalculated. If a sense of word w_i is eliminated, then the frequency information for the remaining senses of w_i will need to be recalculated. This is shown in equation 5.1 where n is the number of senses remaining consideration. If senses of a different word have been eliminated, the maximum and minimum CIS need to be recalculated. This is because the eliminated senses may have contributed towards the CIS scores.

There are two algorithms which are used to eliminate senses at each iteration. The first algorithm is more efficient, but is not available at all stages. The second algorithm is always available, but incorporates a greater chance that a sense which is a member of the best sentence score will be eliminated.

5.3.1 No Intersection Elimination

No Intersection Elimination (NIE) is the more powerful of the two algorithms. When it is available, NIE is able to eliminate many senses from many different words in one iteration. NIE does not compare scores for senses between different words, and therefore operates at the word level. All the senses which are eliminated by this mechanism can not possess a score which is higher than the chosen sense for that word. However, a sense may be eliminated which is a member of the best sentence combination if the eliminated sense enabled other words in the sentence to possess a higher score. A fully worked example taken from SEMCOR is given in section 6.6.1. This example demonstrates an instance where a sense is removed by the NIE method, which had it have been chosen, would have created a higher sentence score.

For each word, the maximum and minimum scores for all possible senses are compared. The NIE algorithm commences by identifying the highest minimum score for each ambiguous word in the sentence. Equation 5.8 shows how the highest minimum score is calculated.

$$Highest\ Minimum\ Score(w_i) = \max\{MinScore_{i1}...MinScore_{im}\} \quad (5.8)$$

where m is the number of senses of w_i. The maximum score for each sense is then compared with the highest minimum score for that word. Any sense with a maximum score less than the highest minimum score is eliminated. This is shown in equation 5.9.

$$MaxScore_{ij} < Highest\ Minimum\ Score(w_i) \Rightarrow eliminate_{ij} \quad (5.9)$$

This algorithm is called No Intersection Elimination, as senses are eliminated when no overlap exists between the possible range of scores of two possible senses of a word. The eliminated sense possesses a lower score than another sense regardless of which senses of the context words are chosen.

In some instances, all of the words possess senses where an intersection exists between the maximum and minimum scores. In these instances the NIE algorithm can not be employed, and this is a drawback of this method. To overcome this problem a second algorithm is required to continue the elimination. The elimination of only one sense by a different algorithm may be sufficient to reduce the level of uncertainty and re-enable the usage of the NIE algorithm.

5.3.2 Normalised Max Score Elimination

The Normalised Max Score Elimination (NMSE) algorithm deals with more difficult instances where no clear evidence exists at the word level regarding which sense should be eliminated. The algorithm aims to minimize the chances that a sense which contributes to the best sentence score is eliminated. To do this NMSE operates at the sentence level, and only one sense is eliminated from the sentence for each iteration. Therefore, the algorithm aims to identify the one sense in the sentence which is least likely to be chosen.

The algorithm considers only the maximum scores for each sense. This is because the senses of the contextual words which are used to achieve the maximum score are more likely to be correct than those used to achieve the minimum score. The maximum scores can not be directly compared between senses of different words as the scores for each word may lie within a different range. This is shown in the example in section 5.4. Therefore, the maximum scores for all senses are normalised by the highest maximum score for each word. The equation used to calculate the normalised score is given in 5.10.

$$Normalised\ Score_{ij} = \frac{MaxScore_{ij}}{\max\{MaxScore_{i1} ... MaxScore_{im}\}} \tag{5.10}$$

The normalised score represents a measure of the strength of each sense relative to the strongest sense for each word. Normalisation of the scores in this way enables the elimination of a sense to be based on the strength of the best sense as well as the weakest sense. For example, if the best sense for a word has a high maximum score relative to the other senses, then that sense is more likely to remain the best sense as more of the uncertainty is removed. Therefore, it is beneficial to eliminate the sense with the lowest score from that word. The NMSE algorithm eliminates the sense which has the lowest normalised maximum score out of all the possible senses in the sentence.

5.4 Example

The process by which possible senses are eliminated from a sentence is now further explained with the aid of an example. To maintain clarity, the example considers a very simple sentence - *"I wear the black suit"*. WordNet assigns twelve senses to *wear*, eleven senses to *black* and eight senses to *suit*. Therefore, despite its shortness, the sentence still possesses 1056 sense combinations. The closed class words *I* and *the* are not considered during the disambiguation process.

Table 5.1: Definitions and frequencies of the senses being considered by the disambiguation example.

Sense	Definition	Freq
wear(1)	To have clothes on	30
wear(2)	To deteriorate or decay	10
black(1)	A colour	20
suit(1)	Type of clothing	15
suit(2)	Part of a pack of playing cards	30

Table 5.2: Contextual scores between senses.

	wear(1)	wear(2)	black(1)	suit(1)	suit(2)
wear(1)	0	4	6	3	2
wear(2)	4	0	6.5	2.5	3.1
black(1)	6	6.5	0	2	1
suit(1)	3	2.5	2	0	1
suit(2)	2	3.1	1	1	0

DURHAM only considers senses belonging to the correct POS which reduces the complexity of the task. In this example DURHAM only considers nine verb senses of *wear*, eight adjective senses of *black* and four noun senses of *suit*. This reduces the number of sense combinations for the sentence to 288. For further simplification, the example commences in the middle of the elimination process, when many of the senses have already been removed. The definitions and frequency information of the remaining senses are shown in table 5.1. The contextual scores between the remaining senses are shown in table 5.2. The values in both tables are not actual values from DURHAM and are used for illustration purposes only.

The elimination process proceeds using the information in tables 5.1 and 5.2 to calculate the maximum score, minimum score and normalised score for each sense. These are calculated using equations 5.6, 5.7 and 5.10 respectively. The values are shown in table 5.3.

Table 5.3 shows that none of the senses can be eliminated by the No Intersection method. For both ambiguous words, the highest minimum score (3.417 for wear and 1.833 for suit) is lower than the lowest maximum score (3.45 for wear and 2.00

Table 5.3: Scores for each sense before any elimination has taken place.

Word	Frequency	Max_CS	Min_CS	Max Score	Min Score	Normalised Max Score
wear(1)	0.75	3	2.667	3.75	3.417	1
wear(2)	0.25	3.2	3	3.45	3.25	0.92
black(1)	1	2.833	2.333	3.833	3.333	1
suit(1)	0.333	1.667	1.5	2.00	1.833	0.983
suit(2)	0.667	1.367	1	2.034	1.667	1

Table 5.4: Scores for each sense after wear(2) has been eliminated.

Word	Frequency	Max_CS	Min_CS	Max Score	Min Score	Normalised Score
wear(1)	1	3	2.667	4.0	3.667	1
black(1)	1	2.667	2.333	3.667	3.333	1
suit(1)	0.333	1.667	1.667	2.0	2.0	1
suit(2)	0.667	1	1	1.667	1.667	0.833

for suit). Therefore, the No Intersection Elimination algorithm can not be adopted at this stage. As a result, the Normalised Max Score Elimination algorithm must be used. The sense with the lowest normalised score is eliminated. Table 5.3 shows that wear(2) has the lowest normalised score (0.92) and is therefore eliminated from consideration.

The example highlights the importance of normalising the scores so that senses from different words can be compared. The range of scores for both senses of wear are much higher than the range of scores for both senses of suit.

Once wear(2) has been removed, the scores are recalculated considering only the remaining possible senses. The recalculated scores are shown in Table 5.4. Note the changed frequency score for wear(1) due to the instances of wear(2) no longer being considered. The contextual scores for black and suit have also changed as wear(2) can no longer provide contextual information.

At this stage only multiple senses for suit remain. The ambiguity of the context words has been resolved. As a result, the max and min scores are the same for both senses of suit. This ensures that the NIE algorithm can be applied. Table 5.4 shows

Table 5.5: Scores of possible sense combinations for the sentence *"I wear the black suit"*

Wear		Black		Suit		Sentence
Sense	Score	Sense	Score	Sense	Score	Score
1	3.75	1	3.67	1	2.00	**9.42**
1	3.42	1	3.33	2	1.67	**8.42**
2	3.25	1	3.83	1	1.83	**8.91**
2	3.45	1	3.50	2	2.04	**8.99**

that the max score for suit(2) is 1.667, which is lower than the min score for suit(1) (2.0). Therefore, suit(2) can be eliminated. Disambiguation is now complete as no words exist with more than one possible sense.

This example highlights the importance of removing the least likely senses at any one time. By so doing, it prevents unlikely senses providing unhelpful contextual information, which could cause an incorrect classification of other words. In this example suit(2) initially had a higher max score than suit(1). However, the high contextual score suit(2) had with the incorrect sense of wear was the reason for this. Once this sense of wear was eliminated, suit(1) maintained the highest score.

The role of the elimination algorithm is to find the sense combination with as high a sentence score as possible. If the elimination algorithm is able to perform this it has fulfilled its role. If the chosen senses belong to the sense combination with the best sentence score and these senses are incorrect, then the cause of the error is the score assigned by the knowledge sources. Table 5.5 shows that for the example, the elimination algorithm correctly chooses the senses which generates the sense combination with the best sentence score.

5.5 Discussion of Elimination Mechanism

The example has shown one of the advantages of only eliminating the least likely sense or senses at any one time. It enables only the remaining senses to provide

contextual information which are the most likely to be correct. However, there is an additional advantage. By eliminating senses one at a time, the framework is set to allow a lazy evaluation. For some NLP tasks, it may not be necessary to resolve the ambiguity of a word completely. For such NLP tasks, the elimination process can be halted at any stage once sufficient ambiguity has been resolved. The NLP system may be designed to carry some of the word ambiguity through to a later processing stage. This could also be accommodated by the elimination process which could be halted to produce an N-best list of possible sense combinations for the sentence.

The mechanism for eliminating senses from a sentence adopted by DURHAM is not claimed to be optimal. It is merely put forward as an alternative approach. Currently it is not possible to compare the methods used to select the correct senses which are incorporated by different systems. Many systems are evaluated on text where only one word is considered ambiguous. For these systems, the correct sense of context words is not considered. This makes the selection process less complex as maximum and minimum scores do not need to be considered. Section 6.6 examines the effect of choosing the correct context sense. Section 6.6 also puts forward some evaluation metrics to test the effectiveness of the selection process.

In [Agirre and Rigau, 1996] all the nouns in a context window are disambiguated. Therefore, consideration of the ambiguity of context words is necessary. Their approach is to resolve the words in sequence. The word W to be disambiguated is in the middle of the context window. All words prior to W will have previously been disambiguated leaving only one sense needing to be considered. All senses of the context words after W will provide equal contextual evidence. Should a sense provide useful evidence which is later not chosen, no back tracking mechanism is proposed.

The most similar method to that used by DURHAM is adopted in [Cowie *et al.*, 1992] and developed further in [Stevenson, 1999]. In these approaches a simulated annealing search mechanism is adopted. Similar to DURHAM, it aims to find the sense combination which produces the highest score. Compared to DURHAM's

approach, simulated annealing is a more random search algorithm which enables the escape from local maxima. By doing so some back tracking is enabled which is not possible in DURHAM's algorithm. The benefit of DURHAM's approach is that individual scores for each sense are considered. These individual scores help direct the search towards the best sentence score. The simulated annealing mechanism is only directed by the total sentence score so only a small percentage of the total number of sense combinations can be considered.

5.6 Summary

To summarise, the disambiguation system obtains scores from two different knowledge sources: frequency, and contextual information. These scores are combined producing a maximum and minimum score for each sense depending on the chosen sense for the contextual words. These scores are then used to iteratively eliminate senses by the elimination of the least likely senses at any one time. This process is continued until all the word level ambiguity has been resolved.

The algorithm introduces a new compromise between considering words in isolation from the sense choices made for the context words and considering all the possible sense combinations for a sentence. The compromise must weigh up the relative importance of efficiency and accuracy. Considering the problem at the word level is more efficient and considering the problem at the sentence level is more accurate. The compromise made in this algorithm slightly favours the word level approach. This seems appropriate for a system in which one of the knowledge sources used is not affected by the sense choices made for the context words.

Chapter 6

Evaluation on SemCor

The previous two chapters have examined the various components which constitute the DURHAM system. This chapter details the performance of this system when it is evaluated on SEMCOR. SEMCOR is the largest widely available large scale sense tagged corpus, and is sense tagged using the WordNet lexicon.

For each sense tagged word in SEMCOR, the POS and the root form of that word is given. This information is used by DURHAM so that only the senses with the correct word form and of the correct POS are considered. The reason for this is that in a typical NLP system the morphology module identifies the root form of a word and the POS tagger identifies the POS. Therefore, these tasks are considered out of the scope of WSD. Knowing the root form of the word does remove a small amount of word level ambiguity. For example, the word *won* could refer to the past tense of *win*, or it could refer to the Korean monetary unit. However, by considering senses of different POS, the number of possible senses for each word is greatly increased. This makes the learning process unnecessarily more complex when the POS of a word can be identified with high accuracy by a POS tagger [Brill, 1992], [Brill, 1995]. Therefore, the task being evaluated in this section is:

Given a list of senses belonging to the correct word form and the correct POS of a particular word, to select the correct sense for that word from the list.

This evaluation is a large scale task, as all open class words in SEMCOR are

disambiguated.

Before any results can be reported, the evaluation metrics adopted must be detailed. Several evaluation metrics are considered in order that various components of DURHAM can be analysed. The chapter then proceeds by detailing the nature of the training and test sets used for evaluation. The results achieved by DURHAM are then reported. This is followed by an analysis of various components of DURHAM which have been discussed in the previous chapters.

An attempt to compare this work with other systems is then carried out. This is a complex task due to the problems of evaluation on different data sets. These problems were detailed in section 2.5. However, a comparison is made with one other system, also evaluated on SEMCOR. Due to the difficulties of comparing work on different data sets, various metrics have been proposed which indicate the complexity of the data set in terms of its ambiguity. These metrics are investigated to determine their correlation with the accuracy achieved by DURHAM.

6.1 Evaluation Metrics

Prior to the results for DURHAM being reported, the evaluation metrics used to assess the performance of DURHAM need to be detailed. Chapter 2 highlighted the lack of standards for evaluation metrics within WSD. As a result, some of the evaluation metrics introduced in this section are unique. However, the metrics aim to be sufficiently thorough that the results can be compared, should any other work be carried out on the same blind test set. For all the evaluation metrics, the sense which has been chosen by the sense tagger is assumed to be correct. Although the ITA agreement may imply that some of the sense tags are incorrect, this assumption needs to be made as there is no way of knowing which senses are misclassified. There are 666 instances in SEMCOR where more than one sense for a word has been assigned. In these instances, DURHAM need only choose any one of the possible senses to be considered correct. The evaluation metrics are now detailed individually.

6.1.1 Random Baseline

The random baseline is the accuracy of a system which always chooses a sense of a word at random. This baseline figure gives an indication for the complexity of the task. This is a particularly useful baseline figure for systems which do not take advantage of training data.

6.1.2 Frequency Baseline

The frequency baseline is a standard bench mark from which results can be measured for systems which do take advantage of training data. The frequency baseline is the accuracy achieved by the system if the most frequent sense is always chosen for each ambiguous word. The frequency information is calculated from the data used to train the system. As DURHAM takes advantage of training data, the frequency baseline is a better bench mark to compare results against than the random baseline.

6.1.3 Fine Grained Accuracy

The fine grained accuracy is the percentage of senses correctly chosen by DURHAM out of all the ambiguous words. The fine grained accuracy is the metric which can be used to compare the DURHAM system with other systems evaluated on the same blind test set.

$$Fine\ Grained\ Accuracy = \frac{Number\ of\ correctly\ resolved\ ambiguous\ words}{Total\ number\ of\ ambiguous\ words} \tag{6.1}$$

6.1.4 Contextual Level Accuracy

The contextual level accuracy metric is designed to evaluate the performance of the contextual information knowledge source. The contextual information generally operates above the word level. In some instances more than one sense of a word is represented by the same node in the contextual matrix. If the node in the contextual matrix which represents the correct sense is selected, the contextual information has fulfilled its role even if the incorrect sense is finally chosen. Therefore, the contextual level accuracy considers a sense choice to be correct if it is represented by the same node in the contextual matrix as the correct sense.

$$Contextual\ Level\ Accuracy = \frac{Number\ of\ correct\ contextual\ matrix\ nodes}{Total\ number\ of\ ambiguous\ words} \tag{6.2}$$

6.1.5 Lex File Accuracy

Within WordNet each sense is assigned one of 45 lexicographer files based on the syntactic category and logical groupings. There are 26 categories for nouns, 15 categories for verbs, 3 categories for adjectives and 1 category for adverbs. Achieving high lex file accuracy is more difficult for nouns and verbs compared with adjectives and adverbs, as there are many more possible categories. The lex file accuracy considers the instances where the chosen sense is of the correct lex file. DURHAM has not been designed to achieve high accuracy at this very coarse level of granularity because it may be to the detriment of the finer grained accuracy. For example, consider three noun senses of the word *bank* shown in table 6.1 along with their lex file groupings and frequency information. For the purposes of the example, the frequency information has not been taken from the SEMCOR training data.

If a system based solely on frequency information aimed to achieve high accuracy at the fine grained level, the system would choose sense 3 in all instances. An accuracy of 40% would be achieved at both the fine grained and lex file level. How-

Table 6.1: The definitions and frequencies for three senses of *bank* being considered

Sense Number	Definition	Lex File	Frequency
1	The side of a river	object	30
2	Building offering financial services	object	30
3	Funds held by a gaming house	possession	40

ever, a system aiming to achieve high lex file accuracy would choose either sense 1 or 2. In this case it would only achieve 30% fine grained accuracy, but would achieve 60% lex file accuracy. The purpose of including this metric is to enable a comparison should other systems evaluate at this level.

$$Lex\ File\ Accuracy = \frac{Number\ of\ correct\ lex\ files}{Total\ number\ of\ ambiguous\ words} \tag{6.3}$$

6.1.6 All Words Accuracy

All of the accuracy metrics detailed above have only considered ambiguous words in the sentence. The reason for this is that within WSD the ambiguous words represent the non trivial subset of all words for the correct sense to be chosen. By including the trivial monosemous words, the overall accuracy of the system is increased. As a result, the accuracy gains made by changes to the system become smaller and therefore less is shown about the performance of the system. However, it is also important to define accuracy in terms of the words in the sentence. This metric is beneficial for developers outside WSD. For example, a developer of a NLP system is interested in the likelihood of a word being tagged with its correct sense. The difficulty to assign the correct sense is of less interest. The "all words" metric gives a better indication of the importance of resolving word ambiguity for their particular task. Developers may be interested in all of the words in the sentence or just the open class words in the sentence. Both of these metrics are therefore given. The accuracy including these additional trivial words is given at the fine grained level.

$$Open\ Class\ Accuracy = \frac{Number\ of\ correctly\ resolved\ words}{Total\ number\ of\ open\ class\ words} \tag{6.4}$$

$$All\ Words\ Accuracy = \frac{Number\ of\ correctly\ resolved\ words}{Total\ number\ of\ words} \tag{6.5}$$

6.1.7 Kappa

A description of Kappa was given in chapter 2. The aim of Kappa is to assign a score between zero and one which represents where the system lies between a chance system and a perfect system. As training data is available for this work, frequency information is available. Therefore, a system achieving chance accuracy is considered to be one which achieves the frequency baseline. As a result, Kappa is calculated as follows for this work.

$$Kappa = \frac{a - freq\ baseline}{1 - freq\ baseline} \tag{6.6}$$

where a is the accuracy achieved by DURHAM.

6.1.8 UBAKappa

An additional formula for calculating Kappa is also proposed. This separate metric takes into consideration a realistic upper bound for that particular data set as well as the lower bound frequency baseline already considered.

In equation 6.6, the 1 in the denominator represents the accuracy of a perfect system. However, it is questionable whether an automatic system is able to achieve 100% accuracy on a data set which contains inter-tagger disagreement. A realistic upper bound for any WSD system is to achieve the same level of accuracy as a human could achieve [Wilks, 2000]. This upper bound is shown as the ITA for that data set. The proposed metric relates the accuracy of the system to both an

upper and lower bound. In this work, this metric will be called UBAKappa (Upper Bound Adjusted Kappa) and is defined as follows.

$$UBAKappa = \frac{a - freq\ baseline}{ITA - freq\ baseline} \tag{6.7}$$

where a is the accuracy of the system. For both Kappa measurements, the accuracy considered will be the fine grained accuracy.

Using this metric, DURHAM will score a UBAKappa value of one if it achieves the same accuracy as a human could achieve on SEMCOR. The entire SEMCOR corpus has not been sense tagged by more than one person to produce a definitive ITA figure. However, [Fellbaum, 1996] report that unskilled lexicographers agreed with the senses assigned in SEMCOR in 74% of all instances. This figure is lower than could be achieved on SEMCOR because unskilled lexicographers perform the sense tagging. However, it is substantially better than the 57% ITA found between the text contained in both SEMCOR and the DSO corpus [Hwee Tou Ng and Hian Beng Lee, 1996]. For this work, it is considered acceptable to set an upper bound for an automatic system to perform comparably with unskilled lexicographers. Therefore, 74% will be used as the upper bound figure for calculating UBAKappa for evaluation on SEMCOR.

6.1.9 Statistical Significance

Analysis of DURHAM will provide results highlighting the effects that different components have made to the overall accuracy of DURHAM. It is insufficient to only examine the change in accuracy in order to be sure that a particular component has made a positive contribution. Additionally it is necessary to test if any improvement made is statistically significant. The significance test chosen to perform this function is called the *The McNemar test for the significance of changes* [Siegel, 1956].

To be able to use this statistical test, the test set must be the same before and

Table 6.2: Fourfold table to represent the way the classification of senses has changed

		After	
		Correct	Incorrect
Before	Incorrect	A	B
	Correct	C	D

after the change has been made. For each word, the test identifies if the chosen sense is correct or incorrect before and after the change. An alternative approach would be to consider each SEMCOR file as a separate sample and examine the difference in accuracy for each file. This approach would apply a t test to the difference scores [Siegel, 1956]. However, by considering each ambiguous word in the test set individually, the sample size is greatly increased enabling a more accurate statistical measure.

The McNemar test establishes a fourfold table of frequencies in order to represent the results before and after the change to the system. The frequencies represent the four possibilities in which a word can be classified, and is shown in table 6.2.

The cases which have shown changes between the before and after test appear in cells A and D. If a word is correctly classified in both test runs it will appear in cell C. If a word is incorrectly classified in both test runs it will appear in cell B. As the statistical test is concerned with how the system has changed only the frequencies from cells A and D are considered.

The formula to calculate the McNemar test is derived from the χ^2 test shown in equation 6.8

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \tag{6.8}$$

where O_i is the observed number of cases in the ith category, E_i is the expected number of cases in the ith category under the null hypothesis (H_0) and k is the number of categories.

Table 6.3: Table showing the probability of H_0 for different values of χ^2

$P(H_0)$	0.15	0.10	0.05	0.025	0.01	0.005	0.0005
χ^2	1.07	1.64	2.71	3.84	5.41	6.64	10.83

The null hypothesis states that any change is by chance and therefore it would expect A and D to be equal. Therefore, the expected number of cases in categories A and D is $\frac{1}{2}(A+D)$. Substituting this value for E_i and applying a correction for continuity generates equation 6.9. This equation constitutes the McNemar test.

$$\chi^2 = \frac{(|a-d|-1)^2}{a+d} \tag{6.9}$$

The region of rejection of H_0 is one tailed, as the statistical significance is only of concern if the changed system performs better than the original and thus $A > D$. Table 6.3 shows the probability of H_0 for different values of χ^2.

6.2 Training and Test data

The SEMCOR corpus consists of 186 sense tagged files, with each file containing approximately 100 sentences. For this work SEMCOR is split into three sections, training data, validation data and blind test data. The training data consists of a section of SEMCOR known as Brown 1. The validation data and blind test data are both taken from Brown 2. Splitting the corpus in this way increases the chances of other systems being evaluated on the same test set in the future and is the reason for this choice. Table 6.4 shows the relative sizes of each of the three data sets. A list of the files used in each data set can be found in appendix A. The purpose of each data set is detailed in chapter 5.

Table 6.4 shows that approximately 55% of SEMCOR is used to train the contextual matrix and derive the frequency information, and a further 17% is used as validation data for this training process. Approximately 28% of SEMCOR is used to evaluate the performance of DURHAM on unseen data. In addition table

Table 6.4: Table showing the size of the training, validation and blind test data sets

	Training Data	Test Data	Blind Test Data
Number of files	103	30	53
Number of sentences	11,182	3,222	5,734
Number of words	198,796	58,000	102,936
Number of open class words	106,639	31,404	54,596
Number of ambiguous word occurrences	82,325	24,059	43,339

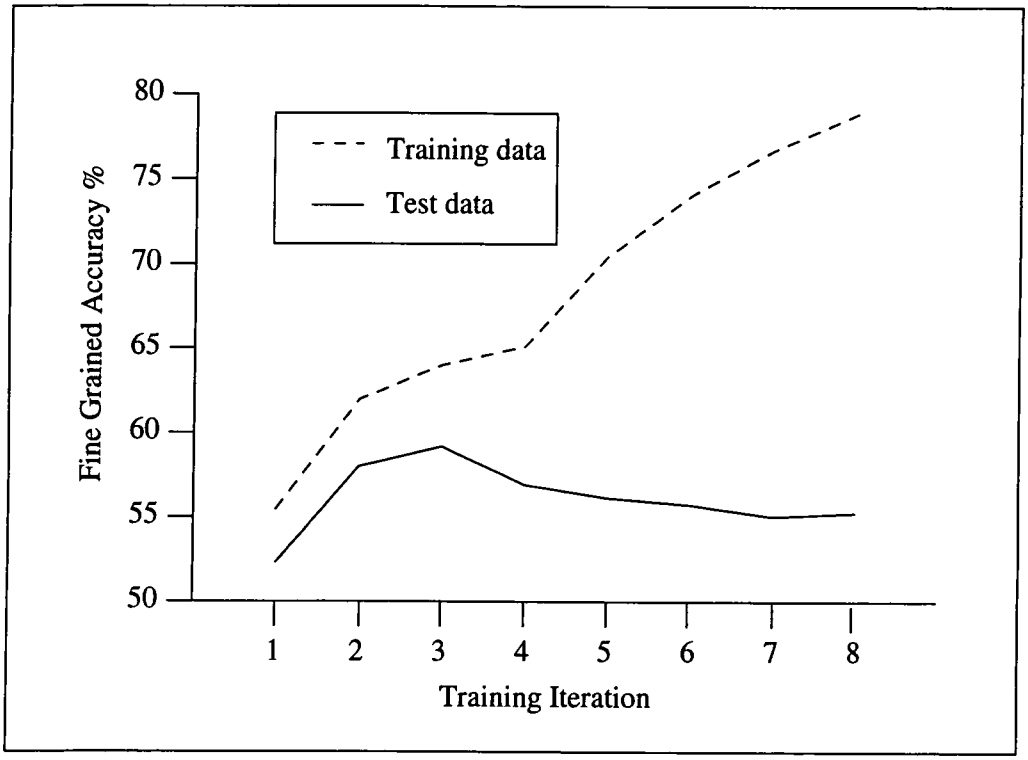
6.4 highlights the relative frequency of open and closed class words in free text. Despite the existence of far more open than closed class words in the lexicon, open class words constitute only 54% of all words in free text. However, the table also shows that 78% of all open class word occurrences are ambiguous.

6.3 Results

This section considers the results achieved by DURHAM on the blind test data. Section 5.2 detailed the method by which knowledge sources are combined and explained how constants are used to weight the contribution of each knowledge source. In the results reported in this section, these constants are both arbitrarily set to 0.5. An analysis of the effect of using different weights is considered in section 6.4.

DURHAM learns contextual scores by examining the training text. The validation data is used to signal an end to the training process. When accuracy on the validation set falls, no further training iterations will be carried out. This method was discussed in section 4.4. Figure 6.1 examines how the accuracy on both the training and validation data changes after each iteration of training. Figure 6.1 shows that after three iterations of the training procedure the highest accuracy is achieved on the validation data. Further iterations of training cause the accuracy on the validation data to decrease. However, the accuracy continues to increase on

Figure 6.1: Graph showing accuracy on validation and training data after each training iteration



the training data set. The accuracy on the training set is 15% higher after eight iterations than it is after three iterations. The reason for this is that the training process is no longer able to make generalised adjustments to the contextual scores to improve accuracy. As a result, the training process makes specific adjustments which are suitable only for the training data. Training which only causes an improvement on seen data is a common problem in other areas of artificial intelligence such as neural networks. The problem is referred to as **Overfitting** [Winston, 1992].

Once the system has finished training, DURHAM is evaluated at the different levels of granularity using the metrics detailed in section 6.1. The results are shown in table 6.5. The results show that DURHAM achieved 62.14% accuracy at the fine grained level. This is slightly under an 11% increase above the frequency baseline.

A greater increase of almost 12% is observed at the contextual level. The reason for this is that the training process alters scores between nodes found in the contextual matrix. Therefore, the training process is more suited to the contextual

Table 6.5: Table showing the results achieved by DURHAM when evaluated on the blind test data. The figures are expressed as a percentage

	Fine	Contextual	Lex file	Open Class	All words
Random baseline	27.70	34.19	71.43	42.61	69.56
Frequency baseline	51.27	55.09	81.56	61.32	79.48
DURHAM	62.14	66.92	86.19	69.94	84.06

level evaluation. However, there is only a small difference in accuracy between the fine grained and contextual level results. This suggests that the contextual matrix extends to a sufficient depth in WordNet to distinguish between all senses in most instances.

The results show that using the training data to provide frequency information is extremely beneficial. The frequency baseline is 23.57% above the random baseline. This is a substantial increase, which suggests that most words have a skewed frequency distribution of senses. If all words had a uniform frequency distribution of senses the frequency baseline and the random baseline would be the same. In addition, the results display a further benefit through using frequency information. Unlike the contextual information, frequency information operates at the fine grained sense level. Therefore, only the frequency information is available to distinguish between two senses which are represented by the same node in the contextual matrix. This benefit can be measured by examining the probability of identifying the correct sense given that the correct node in the contextual matrix is chosen. This is calculated using Bayes theorem.

$$P(Fine|Contextual) = \frac{P(Fine \cap Contextual)}{P(Contextual)} \tag{6.10}$$

Using the results in table 6.5, a random system will correctly identify the correct sense in 81% of all instances if the correct contextual node is given. However, using frequency information increases this figure to 93%.

High accuracy of 86.18% is achieved at the lex file level. However, this is only 4.63% above the frequency baseline. This shows that the lex files are not suited to

identifying semantic categories which separate various senses of a word. This adds evidence to the argument given in section 4.3 which highlighted that all senses of *doctor* were assigned the same lex file category.

The lex file accuracy is reported because it is a level of granularity which can be easily evaluated. By reporting these results the chance of future work being compared with these results is increased. However, the lex file accuracy is reported with some caution. The objectives of DURHAM were to perform well at a finer level of granularity than the lex files. This is reflected by both knowledge sources operating at a finer level of granularity. Therefore, the lex file accuracy can be a misleading metric to assess the performance of DURHAM. It is likely that higher lex file accuracy could be achieved if training and the disambiguation algorithm were targeted at the lex file level.

The "open class" metric considers all open class words, even if the word only possesses one sense and can be resolved trivially. By including these trivial words only a 7% increase in accuracy is generated for the DURHAM results. The reason for this is that most open class words are ambiguous. The figure achieved of slightly less than 70% is useful for a NLP developer to give an indication of the overall accuracy of the disambiguated text.

The "all words" metric also considers closed class words, none of which are ambiguous. As this almost doubles the number of words being considered the accuracy is greatly increased. The DURHAM accuracy for all words at the fine level of granularity is 84.06%. This figure is useful to gain an overall perspective on the WSD problem. However, a difference of only 4.58% between the frequency baseline and DURHAM highlights the reason why the "all words" metric is not suited for comparing different WSD systems.

Using table 6.5 the Kappa values for DURHAM can be calculated as follows:

$$Kappa = \frac{0.6214-0.5127}{1-0.5127} = \mathbf{0.2230}$$

$$UBAKappa = \frac{0.6214-0.5127}{0.74-0.5127} = \mathbf{0.4781}$$

Kappa provides an excellent metric for comparing WSD systems and for highlighting the extent of development still required in WSD. The Kappa value shows that DURHAM is slightly over one fifth of the way towards a perfect system. The UBAKappa value is a little more encouraging. It suggests that DURHAM is just under half the way towards achieving a system which could perform comparably with an untrained human.

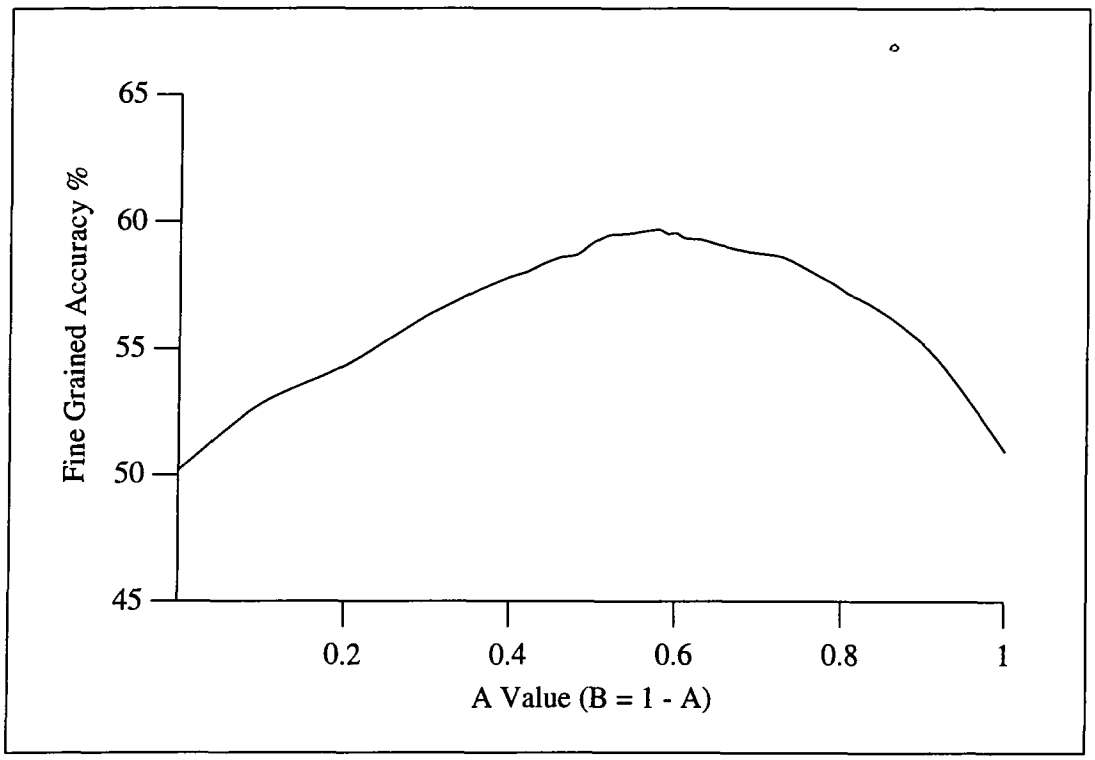
The increase in accuracy above the frequency baseline achieved by DURHAM is statistically significant. Due to the large size of the data set, only small increases in accuracy are required for the improvement to be statistically significant at a 99% level. Table 6.3 shows that χ^2 must be greater than 10.83 for the improved accuracy to be statistically significant at the 99.95% level. The calculated χ^2 for the improved accuracy achieved by DURHAM over the frequency baseline is 1852

6.4 Analysing Constants A and B

Now that the overall results of DURHAM have been reported, various components will be analysed to measure their effect on accuracy. Constants A and B are used in equations 5.6 and 5.7. Their purpose is to weight the contribution each knowledge source makes to the overall score for each sense. This section examines the effect that varying the constants A and B has on the accuracy of DURHAM. This is carried out in order to see if an improvement in accuracy can be made on the blind test data. In addition the relative importance of each knowledge source can be examined.

Analysing A and B values can not be performed using the training data. The best values for A and B on the training data will not necessarily be the same as on unseen data. The reason for this is that the training data has been used to derive the frequency information and train the contextual matrix. Optimum values for A and B on the training set would not represent the ability of each knowledge source

Figure 6.2: Graph showing the effect different values of A and B have on fine grained accuracy on the validation data



to perform on unseen data. Furthermore, the blind test data can not be used. Finding better values for A and B is considered part of the training phase and can not be carried out on data reserved for testing.

The values for A and B which produce the highest fine grained accuracy are found on the validation data. This is performed by iterating through different values of A and B. For all values considered, the sum of A and B is equal to one. The results are shown in figure 6.2.

The results show that the best values for A and B on the validation data are $A = 0.58$ and $B = 0.42$. Applying these constants to the evaluation on the blind test data leads to an increased achieved accuracy of 62.29%. However, this increase is not statistically significant even at the 0.1 level.

Table 6.6: Table showing statistical significance at the 95% level for constant A in the range 0.52 - 0.70. B = 1 - A

	0.58	0.56	0.60	0.54	0.52	0.62	0.64	0.50	0.66	0.68	0.70
0.58	-	X		X							
0.56	X	-	X	X	X						
0.60		X	-	X	X						
0.54	X	X	X	-	X	X	X				
0.52		X	X	X	-	X	X				
0.62				X	X	-	X	X			
0.64				X	X	X	-	X			
0.50						X	X	-			
0.66									-		
0.68										-	
0.70											-

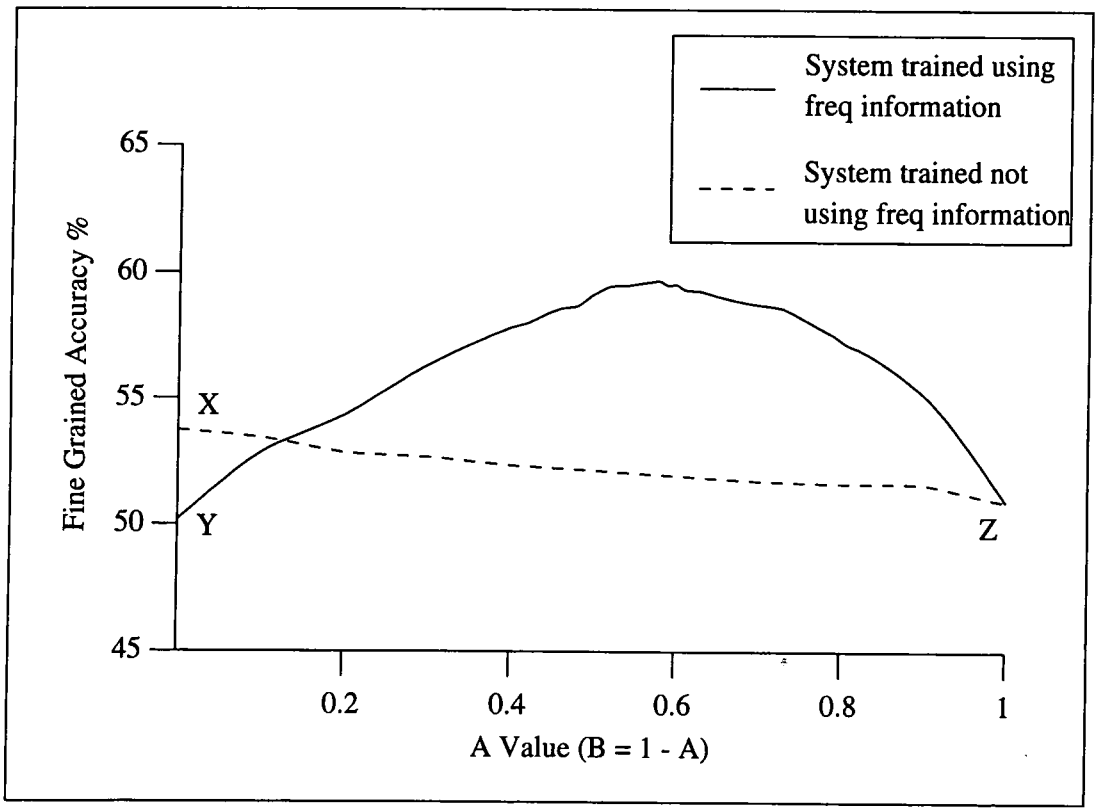
6.5 The Role of Frequency Information

Many benefits of using frequency information have already been highlighted in section 6.3. The frequency information has been shown to substantially increase accuracy particularly at the fine grained level. This section investigates a further benefit of the frequency information.

During the training of the contextual scores, DURHAM disambiguates each training sentence based on the information it possess. Frequency and contextual information are combined to perform the disambiguation. This section investigates whether better contextual scores would be learnt if only the contextual information was used during the training of contextual scores, the principle being to train each knowledge source in isolation and then determine the best way for them to be combined.

The system was re-trained on the same training data. The A and B constants used to weight the contribution from each knowledge source were set to zero and one respectively in order that no frequency information was considered. Five iterations of training were performed before accuracy on the validation data signalled an end to the training process. The new system was then tested on the validation data using different values of constants A and B. The results achieved are compared

Figure 6.3: Graph showing the effect using frequency information during the training of contextual scores makes to the accuracy of the system.



with the original system in figure 6.3.

Figure 6.3 shows that training the contextual scores in isolation from frequency information achieves higher accuracy when the frequency information is also not used during testing. However, the figure also indicates that this method does not achieve such high accuracy overall. The combined knowledge is not greater than the individual knowledge sources. Further analysis highlights the reason for this.

Three different systems are compared:

- System X: No frequency information is used in either training or testing of the system.
- System Y: Frequency information is used during training of the contextual scores, but is not used during testing.
- System Z: A system tested using frequency information only.

Each of the three systems is marked in figure 6.3. System X achieved 53.73% accuracy on the validation data, which is 3.61% higher than system Y. However, 93.4% of all the correct answers given by system X were also correct in system Z. Only 67.8% of all correct answers given by system Y were also correct in system Z.

The results show that system X is similar to system Z. By not incorporating the frequency information into the training of the contextual scores, the contextual information will learn to choose the most frequent sense, and is the reason for the similarity. Thus the contextual information and the frequency information become very similar and are unable to complement each other. By contrast, incorporating the frequency information into the training process helps direct the training of the contextual information. This enables it to learn features in the sentence which may contribute to a sense other than the most frequent being chosen.

6.6 Analysis of the Disambiguation Algorithm

The previous sections have analysed the performance of DURHAM's knowledge sources, the role they perform and the way they are combined. This section progresses on from that analysis and considers the performance of the disambiguation algorithm. The disambiguation algorithm is discussed in chapter 5, and its purpose is to select a sense for each ambiguous word based on the information given from the knowledge sources.

The disambiguation algorithm aims to identify the sense combination which generates as high a sentence score as possible for each sentence. However, due to the large number of sense combinations, particularly for long sentences, this is a difficult task. In order to evaluate the performance of the disambiguation algorithm, the score of the chosen sense combination must be calculated for each sentence. However, the chosen sentence score can not be compared against the best sentence score because the best sentence score can not be calculated efficiently.

Instead the chosen sentence score is compared against the correct sentence score (the score of a sentence with all the correct sense choices). By comparing these

scores, the error can be catagorized into two types. This first type is error caused by the knowledge sources assigning incorrect scores. The second type is the error caused by the disambiguation algorithm failing to identify a sense combination with a higher score.

In the blind test set there are 5734 sentences. For 4516 of these sentences the correct sentence score is lower than the chosen sentence score. Therefore, the scores from the knowledge sources are the cause of the misclassifications. 575 sentences have a higher correct sentence than chosen sentence score and therefore for these sentences the disambiguation algorithm could have done better. The correct sentence is chosen for the remaining 643 sentences.

These results show that the knowledge sources cause the incorrect choice of sense combination in 78.8% of all the sentences. The disambiguation algorithm is, at least partly, the cause of the misclassification in 10.0% of all sentences. The results also show that despite achieving 62% accuracy at the word level, only 11.2% of sentences are sense tagged correctly.

6.6.1 NIE and NMSE

Section 5.3 detailed the two elimination algorithms (NIE and NMSE) which combine to form the core of the disambiguation algorithm. This section examines the relative performance of each of these elimination algorithms. To perform this only the 575 sentences were considered. These sentences are those where the disambiguation failed to select a sentence combination whose score is as high as the correct sentence score. The reason for choosing these sentences is that the elimination algorithm could have done better on these sentences. On the 575 sentences a total of 22219 senses were eliminated. 66.8% of the senses were eliminated by the NIE algorithm and the remaining 33.2% were eliminated by the NMSE algorithm. From all the eliminated senses, 1293 were correct senses. 746 of these correct senses were eliminated by the NIE algorithm and the remaining 547 were eliminated by the NMSE algorithm. Therefore, the NIE algorithm incorrectly eliminated a correct sense in 5.0% of all of the senses it eliminated, and the NMSE algorithm incorrectly

eliminated a correct sense in 7.4% of all the senses it eliminated. This difference in accuracy between the two elimination algorithms is to be expected, as the NMSE algorithm deals with more difficult instances.

To gain a greater understanding of the why the elimination algorithm selects a sub optimal sense combination, an example is investigated. The example sentence is taken from SEMCOR and is one of the 575 sentences where the correct sense combination achieved a higher score than the chosen sentence. The sentence is: *"And the USSR existed as the revolutionary experiment in radical socialism, the ultimate exemplar."* and is taken from file br-g21. The example was chosen for simplicity because a low number of sense combinations exist in the sentence. Table 6.7 shows the starting point for the disambiguation algorithm. The senses highlighted in bold are the correct senses for each word. DURHAM disambiguates all the ambiguous words correctly except for *experiment.* However, table 6.7 shows that for *revolutionary* and *radical* the correct sense is not initially the sense with the highest max score. This highlights the importance of an iterative disambiguation procedure.

After some senses have been eliminated the cause of the misclassification of *experiment* can be observed. The remaining senses are shown in table 6.8. The correct sense (*experiment(1)*) possesses a higher maximum and minimum contextual score than the chosen sense - *experiment(3).* However, *experiment(3)* has a much higher frequency score. As a result the minimum score for *experiment(3)* (2.96) is higher than the maximum score for *experiment(1)* (2.89). Therefore, *experiment(1)* is eliminated by the NIE algorithm.

However, *experiment(1)* provides a high contextual score for many of the other words in the sentence. As a result, the increase in the sentence score affected by choosing *experiment(3)* is lost through many of the other words producing a lower score. This is shown in table 6.9 where the scores for the correct and chosen sense combination are compared. The table shows that although the only difference between the two sense combinations, is the sense choice of *experiment*, the scores for all words are different. The chosen sense of *experiment* receives a higher score

Table 6.7: All possible senses for the sentence before any elimination is performed.

Word	Prob	Max_CS	Min_CS	Max Score	Min Score	Normalised Max Score
USSR(1)	1	7.40	5.75	3.69	2.99	1
exist(1)	0.03	6.73	3.83	2.84	1.62	0.78
exist(2)	0.97	7.35	5.83	3.65	3.01	1
revolutionary(1)	0	6.27	4.55	2.63	1.91	0.99
revolutionary(2)	0	6.36	2.53	2.67	1.06	1
revolutionary(3)	0	6.19	4.03	2.60	1.69	0.97
experiment(1)	0.14	6.69	5.52	2.89	2.40	0.90
experiment(2)	0.06	7.42	4.75	3.15	2.03	0.98
experiment(3)	0.80	6.57	5.63	3.22	2.83	1
radical(1)	0	5.29	3.96	2.22	1.66	0.73
radical(2)	0	4.92	3.28	2.07	1.38	0.68
radical(3)	0	6.01	4.03	2.52	1.69	0.83
radical(4)	0	6.02	3.29	2.53	1.38	0.83
radical(5)	0	7.25	2.53	3.04	1.06	1
socialism(1)	0	5.38	3.83	2.26	1.61	0.74
socialism(2)	1	5.91	3.51	3.06	2.056	1
ultimate(1)	0.33	6.47	5.28	2.91	2.41	1
ultimate(2)	0.33	6.07	5.00	2.74	2.29	0.94
ultimate(3)	0.33	4.83	2.28	2.22	1.15	0.76
exemplar(1)	1	6.55	4.72	3.33	2.56	1

Table 6.8: Table showing why the correct sense of *experiment* is eliminated by the NIE algorithm.

Word	Prob	Max_CS	Min_CS	Max Score	Min Score	Normalised Max Score
USSR(1)	1	6.71	5.92	3.40	3.07	1
exist(2)	1	7.22	5.87	3.61	3.04	1
revolutionary(1)	0	5.93	4.84	2.49	2.03	0.96
revolutionary(2)	0	5.58	3.58	2.35	1.50	0.90
revolutionary(3)	0	6.19	5.02	2.60	2.11	1
experiment(1)	0.14	6.69	6.09	**2.89**	2.64	0.90
experiment(2)	0.06	7.42	5.34	3.15	2.28	0.98
experiment(3)	0.80	6.57	5.94	3.22	**2.96**	1
radical(1)	0	4.79	4.29	2.01	1.80	0.74
radical(2)	0	4.92	3.94	2.07	1.65	0.76
radical(3)	0	6.01	5.02	2.52	2.11	0.93
radical(4)	0	6.02	4.47	2.53	1.88	0.93
radical(5)	0	6.47	3.58	2.72	1.50	1
socialism(2)	1	5.91	4.12	3.06	2.31	1
ultimate(1)	0.5	6.47	5.51	3.01	2.60	1
ultimate(2)	0.5	6.07	5.42	2.84	2.57	0.94
exemplar(1)	1	6.55	5.69	3.33	2.97	1

(3.03) than the correct sense (2.89). However, the total sentence score is higher for the correct sense combination.

The example has outlined that the disambiguation algorithm is not fully able to consider how the elimination of a sense will affect the other words in the sentence. This is mainly due to the frequency information being independent of the contextual words.

The example also demonstrates the ability of DURHAM to generalise by the contextual information operating above the word level. The words *revolutionary* and *radical* do not appear in the training data. Therefore, despite neither word possessing any frequency information or having any instances to be trained on, they are both disambiguated correctly.

Table 6.9: Table comparing the correct and chosen scores for each word in the sentence.

Word	Chosen	Correct
ussr	3.19826	3.18776
exist	3.29309	3.27209
revolutionary	2.50455	2.59905
experiment	3.03104	2.89191
radical	2.47389	2.51589
socialism	2.9848	3.0163
ultimate	2.82388	2.88688
exemplar	3.27355	3.31555
TOTAL	23.583	23.6848

6.6.2 The Effect of Correct Context

The above example demonstrated the effect of a different sense choice for *experiment* on the scores for all other words in the sentence. Despite the incorrect sense choice for *experiment* the remaining ambiguity was still able to be resolved correctly. This section examines the importance of identifying the correct sense of the context words.

In order to investigate the importance of accurate context, the contextual score for each word is calculated by considering only one sense for each context word. As a result, the maximum and minimum contextual scores for each sense are always the same making the disambiguation algorithm less complex. Three tests were performed on the blind test data. The first two tests served as lower bound baselines. A random sense and then the most frequent sense was used as the context sense. The third test was an upper bound baseline with the correct sense being used as the context sense. The results are shown in table 6.10. The table shows that the correct sense choice for the context words greatly affects the extent to which the ambiguity of a word can be resolved accurately. The accuracy for all POS is increased which implies that all words are reliant on accurate context. For all POS, DURHAM achieves 4.53% higher accuracy than the most frequent sense baseline. This shows that the more complex disambiguation algorithm adopted by

Table 6.10: Table showing the effect that choosing the correct sense for the context words has on the disambiguation accuracy. The figures are expressed as a percentage

Context Sense Choice	Nouns	Verbs	Adjectives	Adverbs	**Overall**
Random	63.67	46.37	49.66	63.73	**56.06**
Most Frequent	64.90	49.02	49.52	65.84	**57.60**
Correct	76.22	58.50	59.41	70.78	**67.61**
DURHAM	69.36	54.12	54.65	66.23	**62.14**

DURHAM is worth the computational expense. If the correct sense of the context words is always used a 5.48% accuracy improvement over DURHAM can be achieved. This shows that if one word in the sentence is incorrectly disambiguated it can lead to other words also being disambiguated incorrectly due to the less accurate contextual information available.

These results may be beneficial to a NLP developer. The developer may employ other methods which may be able to eliminate some of the possible senses, for example parsing or semantics. By using the WSD module as late in the NLP process as possible the contextual information may be more accurate. This has been shown to be of great benefit to the accuracy with which the ambiguity can be resolved.

6.7 Examining POS

Further analysis of the DURHAM performance can be made by individually evaluating each syntactic category. Fine grained disambiguation accuracy for nouns, verbs, adjectives and adverbs are shown in table 6.11.

The table shows that DURHAM's accuracy is above the frequency baseline for all four syntactic categories. The Kappa values are given so that the accuracy can be compared relative to the frequency baseline. The highest Kappa values are achieved for nouns and adjectives. The smallest increase in accuracy is achieved for verbs. This may be the result of a low random baseline but relatively high

Table 6.11: Table showing the fine grained results for each POS. The accuracy figures and Kappa are expressed as a percentage.

	Nouns	Verbs	Adjective	Adverbs
Number of instances	19874	13189	6897	3379
Random baseline	35.27	15.63	26.98	31.76
Frequency baseline	57.43	48.39	36.10	57.21
DURHAM	69.36	54.12	54.65	66.23
Kappa	28.01	12.13	29.02	21.09

frequency baseline. This shows that verbs possess a large number of senses and a skewed frequency distribution. As a result, it is more difficult to improve the accuracy beyond the frequency baseline.

A possible reason for the smaller improvement in adverbs over nouns and adjectives is the amount of training data available. The table shows that only 7.80% of the ambiguous words in the blind test set are adverbs. A similar proportion can also be expected in the training set. A further reason may be the high frequency baseline for adverbs, particularly in comparison with adjectives. It is easier to make improvements beyond the frequency baseline if the frequency baseline is low. It is for this reason, that adjectives have the highest value for Kappa. This highlights a potential weakness of using the Kappa metric.

6.8 Comparison with Agirre and Rigau

Due to the evaluation problems considered in chapter 2, it is not possible to compare this work with the majority of other WSD systems. A fair comparison can only be performed if the evaluation has taken place on the same data set.

The *interest* corpus [Bruce and Wiebe, 1994] consists of 2,369 sentences each containing the word *interest*. Each instance is sense tagged with one of six LDOCE senses. As only one word is sense tagged by the corpus, the corpus is more suited to evaluating a small scale WSD system. A better analysis of the application of DURHAM to a small scale evaluation can be carried out through SENSEVAL

Table 6.12: Table comparing the fine grained accuracy of DURHAM with Agirre and Rigau's system on four SEMCOR files.

	br-a01	br-b20	br-r05	br-j09	Overall	
					Fine	Lex
Agirre and Rigau	46%	44%	39%	44%	**43%**	**53.9%**
DURHAM	75%	72%	70%	87%	**76%**	**87%**

which is considered in chapter 7. Therefore, DURHAM has not been evaluated on the *interest* data set.

Very few systems have been evaluated on the SEMCOR data set. This is predominantly due to the difficulty of the large scale task. The work reported in [Stevenson, 1999] and [Wilks and Stevenson, 1998] is evaluated on the SEMCOR corpus. However, this system requires the LDOCE dictionary definitions which force Stevenson and Wilks to use mappings between WordNet and LDOCE. These mappings are incomplete and prevents any kind of comparison between results. This is supported in [Paliouras *et al.*, 1999] which aims to compare this work with their evaluation on SEMCOR, which, like Wilks and Stevenson, uses the LDOCE lexicon. They report that no comparison can be made.

Evaluation on SEMCOR using WordNet mappings is reported in [Agirre and Rigau, 1996]. Agirre and Rigau report accuracy on four SEMCOR files - br-a01, br-b20, br-j09 and br-r05. All of these files are contained in the training data set for DURHAM. To enable a fair comparison between the two systems DURHAM is re-trained with these four files removed from the training data set. The four files are not seen by DURHAM before testing commences. In addition, Agirre and Rigau's system attempts only to disambiguate nouns. Despite the capability of DURHAM to disambiguate all open class words regardless of their POS, only the disambiguation accuracy for the nouns can be compared. For the individual files only the fine grained accuracy is reported. However, the overall results can be compared at the fine grained and lex file level. The results for both systems are shown in table 6.12.

The results show that on all four SEMCOR files, DURHAM achieves higher

accuracy than achieved by Agirre and Rigau. Overall DURHAM achieves 33% higher accuracy at the fine grained level and 33.1% higher accuracy at the lex file level.

The main difference between the two systems is that DURHAM takes advantage of training data, but Agirre and Rigau's system does not. Instances may occur where no training data can be provided to train a WSD system. In such instances an approach such as Agirre and Rigau's may be adopted. However, these results show that a substantial increase in accuracy is achievable if training data is available. This may encourage a developer to invest the required resources to produce training data in order to benefit from increased WSD accuracy.

Agirre and Rigau use the WordNet hierarchy to calculate semantic distance measures between concepts. It is unlikely that the formula adopted to calculate the semantic distance measure is the cause of the low disambiguation accuracy. A more likely cause is the structure of the WordNet hierarchy. The WordNet hierarchy was not developed to aid WSD. The results indicate that semantic distance measures obtained using WordNet are not beneficial in producing high disambiguation accuracy. The alternative contextual information proposed here offers a way forward, but does require training data.

6.9 Evaluating Complexity Metrics

The difficulty of comparing different WSD systems was discussed in Chapter 2. This was further highlighted in section 6.8 which demonstrated that this work could only be compared with one other WSD system. As that system did not use training data, even this comparison is not ideal. The main difficulty is that different systems choose to evaluate on data sets which are most appropriate for their work.

In order to move towards overcoming these difficulties, metrics have been established which aim to quantify the WSD difficulty of a particular data set. The aim of these metrics is to enable a comparison of disambiguation accuracy based on the

difficulty of the task. This section identifies four metrics which give an indication of the difficulty of the task.

The random baseline If a test file contains words with many senses it will be more difficult to disambiguate. This is reflected by the accuracy achieved on the test file by choosing a sense at random.

The frequency baseline The difficulty of a test file is dependent on the frequency of the most common sense for each word in the test set. This information is encapsulated by the frequency baseline.

Average Polysemy This metric is used in [Stevenson, 1999]. It finds the average number of senses per word in the text and is calculated as shown in equation 6.11. Unlike the random baseline, average polysemy does not take into consideration the varied number of senses that different words in the text possess.

Average Entropy Entropy was discussed in section 2.5.2 and has been used in [Kilgarrif and Rosenzweig, 2000]. It combines both the number of senses and the frequency distribution of those senses into its metric. Average entropy is calculated using equation 6.12

$$Average\ Polysemy = \frac{\sum_{i=1}^{N} S(i)}{N} \tag{6.11}$$

$$Average\ Entropy = \frac{\sum_{i=1}^{N}(-\sum_{j=1}^{S(i)} p_{ij} log_2 p_{ij})}{N} \tag{6.12}$$

where N is the total number of ambiguous words in the text, $S(i)$ is the number of senses of word i and p_{ij} is the probability of sense j of word i.

The metrics are calculated for each of the 53 blind test files. The success of each metric is measured by its ability to predict the accuracy achieved by DURHAM for each of the 53 files. In order to implement this, the values of each metric are correlated with the accuracy achieved by DURHAM for all of the blind test files. The correlation is determined using bivariant correlation coefficients. These

Table 6.13: Table showing the bivariant correlation coefficients between various complexity metrics.

	Random Baseline	Frequency Baseline	DURHAM	Kappa	Average Entropy	Average Polysemy
Random Baseline	1	**0.406***	0.172	-0.158	-0.139	-0.071
Frequency Baseline	**0.406***	1	0.064	**-0.658***	-0.126	0.272
DURHAM	0.172	0.064	1	**0.697***	-0.107	-0.232
Kappa	-0.158	**-0.658***	**0.697***	1	-0.017	**-0.393***
Average Entropy	-0.139	-0.126	-0.107	-0.017	1	**0.722***
Average Polysemy	-0.071	0.272	-0.232	-0.393	**0.722***	1

* Correlation is significant at the 0.05 level.

coefficients range from -1 to 1. A high negative or positive coefficient represents a strong relationship between the two variables. A coefficient of zero represents no relationship between the two variables. The correlation coefficients between all the metrics are shown in table 6.13.

Table 6.13 shows that none of the four difficulty metrics have a statistically significant relationship at the 0.05 level with the accuracy achieved by DURHAM. The table shows a strong relationship between DURHAM and Kappa. However, Kappa is an alternative way of expressing the system's accuracy and not a metric for evaluating the difficulty of the test set.

Entropy aims to combine the number of possible senses and the frequency distribution of those senses into its measure. However, there is a strong relationship between the average entropy and average polysemy, and a weak relationship between average entropy and the frequency baseline. This suggests that with the entropy measure, the number of possible senses is the more dominant feature. This evidence supports the criticism of entropy given in section 2.5.2. The graphs showing that there is no relationship between the achieved accuracy and either the average entropy or the average polysemy are shown in 6.4 and 6.5.

The correlation co-efficient between the frequency baseline and DURHAM is

Figure 6.4: Graph showing the relationship between the average entropy and achieved accuracy on the 53 blind test files.

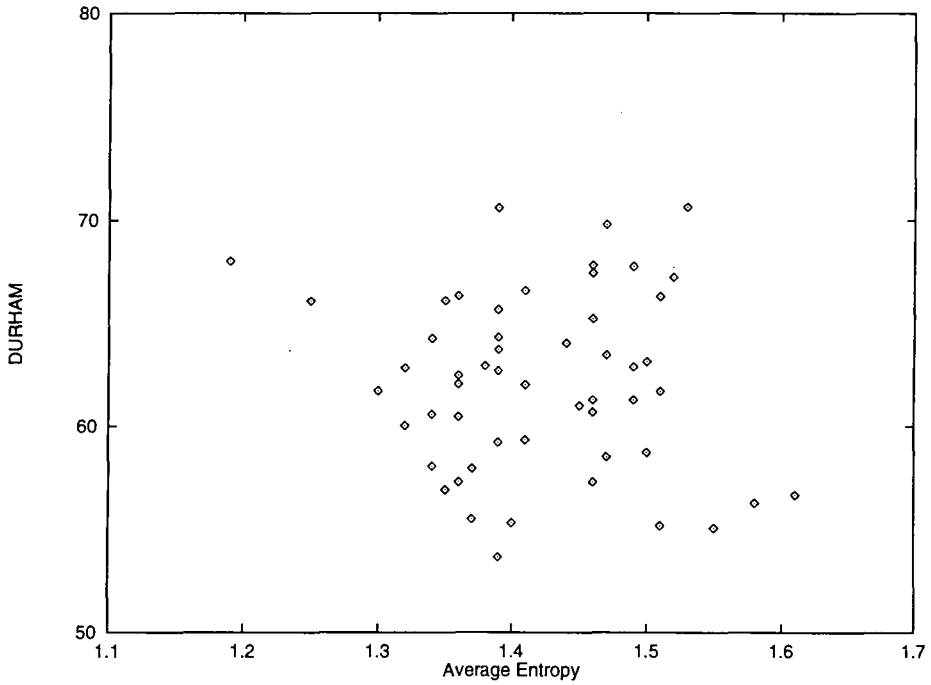


Figure 6.5: Graph showing the relationship between the average polysemy and achieved accuracy on the 53 blind test files.

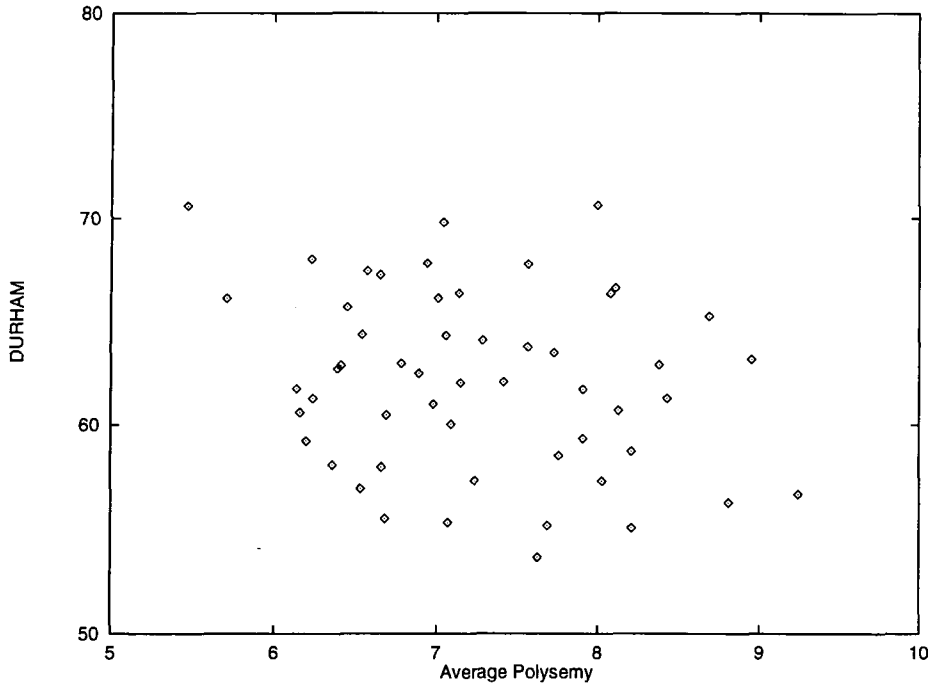
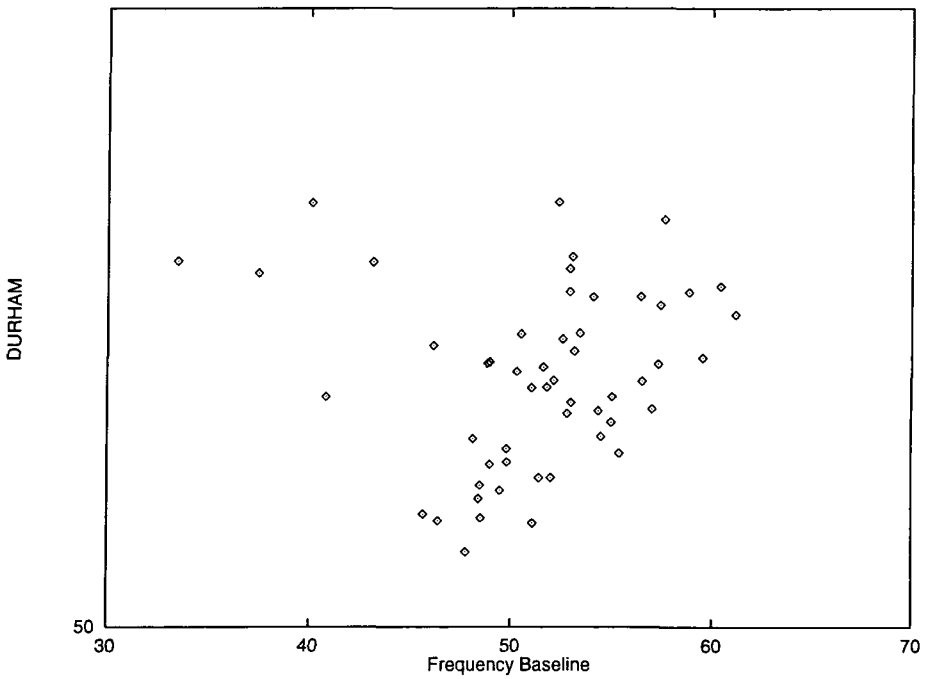


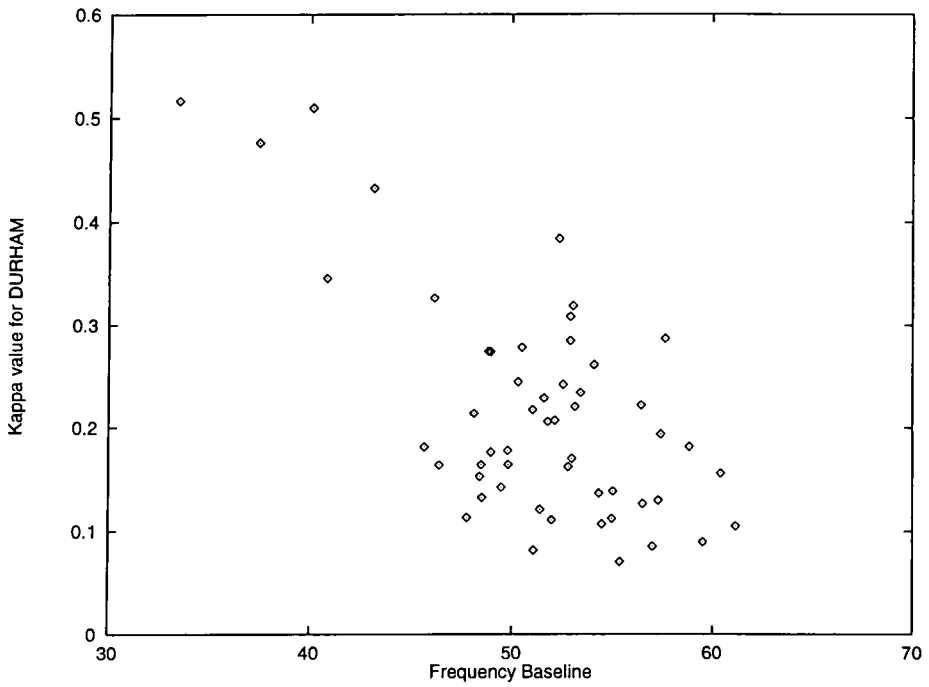
Figure 6.6: Graph showing the relationship between the frequency baseline and achieved accuracy on the 53 blind test files



extremely low suggesting that there is no relationship. However, the graph shown in figure 6.6 indicates that some relationship may exist. Figure 6.6 shows that for most of the blind test files a positive relationship does exist between the frequency baseline and DURHAM. However, five files do not conform to this relationship and thus reduces the correlation co-efficient. By examining the Kappa values, part of the reason for the lack of conformity can be explained. There exists a strong negative relationship between Kappa and the frequency baseline as shown in figure 6.7. This suggests that it is easier to achieve accuracy above the frequency baseline on test files where the frequency baseline is low. The five files in the uppermost left corner of figure 6.7 are the same files which prevent a positive relationship between the frequency baseline and DURHAM in figure 6.6.

These results suggest that none of the metrics investigated give a measure of WSD difficulty that relates to the accuracy achieved by DURHAM. This highlights the importance of evaluating WSD systems on the same data set in order that a fair comparison can be made without the need to consider the difficulty of the task.

Figure 6.7: Graph showing the relationship between the frequency baseline and Kappa on the 53 blind test files



6.10 Summary

This chapter has detailed several evaluation metrics. These evaluation metrics have then been applied to assess the accuracy of DURHAM at disambiguating SEMCOR. The metrics have shown that DURHAM performs well and significantly better than the frequency baseline. Analysis of the two knowledge sources showed that using the frequency information during training of the contextual information enables the knowledge sources to complement each other better. The effect of varying the weight each of the knowledge sources contributes to the overall score was also shown.

An analysis of the disambiguation algorithm was also performed. This demonstrated that correctly sense tagged context words greatly increased the accuracy of the disambiguation system. The elimination algorithms were shown to perform well at identifying sense combinations, which produced a high sentence score.

In addition, this chapter demonstrated that DURHAM fulfilled some of the criteria for success set out in section 2.7. DURHAM has been shown to be large scale.

DURHAM has been able to disambiguate all ambiguous words in real text and is able to achieve high accuracy on them. Also, as much as is possible, DURHAM has shown good accuracy relative to other WSD systems. This is required for the usability criterion. This criterion and the flexibility criterion will be considered further in the following chapter.

Chapter 7

Adaptation to a Small Scale Task

7.1 Introduction

Chapter 6 demonstrated the ability of DURHAM to be applied to a large scale task. The chapter showed that on the large scale task DURHAM achieved good accuracy, but that the accuracy could not be extensively compared with other systems. This chapter aims to examine the accuracy of DURHAM when compared with other systems. This is necessary in order to examine the usability criterion for success of this work. In addition, the flexibility criterion for success will be demonstrated by the ability of DURHAM to be adapted to a different task which uses a different lexicon. The separate WSD task DURHAM is evaluated on is the SENSEVAL evaluation. This enables an extensive comparison of results with other systems.

This chapter proceeds initially by highlighting the differences between the SEMCOR and SENSEVAL evaluation. Each of the major components of DURHAM is then examined, and any adaptations made for the SENSEVAL task are considered. The major adaptation is the inclusion of a new knowledge source named **clue words** and this is discussed in detail. The results achieved on SENSEVAL are presented and compared with the other systems entered into the evaluation. Various components of DURHAM are then examined to analyse the effect they

have on results. Finally, the chapter is concluded with a discussion concerning the scalability of DURHAM and the SENSEVAL evaluation.

7.2 SENSEVAL Evaluation

The SENSEVAL evaluation is discussed in detail in section 2.6. This section summarises the important features of the evaluation and highlights the areas which differ between evaluation on SEMCOR and SENSEVAL. It is important to identify these areas in order to understand how to adapt DURHAM for the SENSEVAL task. These differences are now listed:

- The SENSEVAL task is small scale. Only 35 ambiguous words are considered for disambiguation and no adverbs are considered.
- The ambiguous words are tagged with HECTOR senses for SENSEVAL, but WordNet senses are used in SEMCOR.
- The SENSEVAL task requires only one word per sentence to be disambiguated, whereas all open class words were disambiguated in SEMCOR.
- There are many more training instances per ambiguous word in SENSEVAL. On average 426 training instances were provided per ambiguous word.
- Unlike SEMCOR, the root word form is not provided in SENSEVAL.
- For five words the POS of the word is not given in SENSEVAL.
- For four words no training data is provided in SENSEVAL.
- For all senses in SENSEVAL a complete HECTOR dictionary definition is available.
- Multi word expressions and short phrases are considered additional senses in HECTOR but are classed as separate words in WordNet. For example, *scrap heap* is a sense of scrap in HECTOR but not in WordNet.

Based on the differences between the SEMCOR and SENSEVAL tasks highlighted above, this chapter now examines the adaptations made to the various components of DURHAM.

7.3 Adaptations to the Frequency Knowledge Source

Several of the required adaptations to DURHAM were made by modifying the method in which the frequency information is calculated. This modification is possible due to the large amounts of training data available per word. Large amounts of training data for each word greatly strengthens the value of the frequency information. The identified problem of not being provided with the root forms of the ambiguous words is overcome by this adaptation. Moreover, this adaptation provides more specific frequency information than was previously possible in SEMCOR which increases accuracy.

The modification identifies that information aiding ambiguity resolution is available at the morphology level. Therefore, this information is incorporated into the frequency measure. This is carried out by calculating the frequency distribution of senses for all the word forms, rather than simply using the root form frequency distribution. As the frequency distribution of senses can be very different for differing word forms, the word form frequency distribution provides useful information making the frequency information more specific to each individual example. A word form may appear in the test set, which has not appeared sufficiently frequently in the training set to obtain accurate frequency information. In these instances the root form frequencies are used.

Even in cases where sufficient training data is available, the morphology information is often overlooked by WSD developers. The reason for this is that dictionary entries only exist for the root form of each word. Many systems immediately convert all words to their root form so that they can be identified in a lexicon, and thus lose the word form information.

Table 7.1: Table showing the frequency information for noun senses of *promise* in different word forms.

Sense	Definition	Promise	Promises	Root Form
537566	Declaration to do something	58.7%	74.9%	64%
537626	Showing potential	28.9%	0.6%	19.6%
538409	"To keep one's promise"	5.8%	7.3%	6.3%
538411	"Can't make any promises"	0.0%	17.3%	5.7%
537573	Something will come about	6.6%	0.0%	4.4%

The following example shows how the morphological information can be beneficial for choosing the correct sense. The example uses the five most frequently used noun senses of the word *promise*, which is one of the words considered in SENSEVAL. Table 7.1 gives the probability of each sense appearing in word forms *promise* and *promises*. The probabilities are also given if only the root forms are considered. The probabilities have been determined from the SENSEVAL training data.

Table 7.1 outlines how the word form information is particularly beneficial for identifying specific information assisting with the acquisition of an accurate probability measure for the *showing potential* sense of promise (537626). If the word found in the text is *promise*, there is a 28.9% chance that the sense is referring to the *showing potential* meaning. However, if *promises* is found the probability of the same sense is only 0.6%. It is interesting to note that in this example the frequency baseline is unaffected regardless of whether root form or word form frequencies are used. This is because, for all word forms considered the *declaration to do something* sense (537566) is the most frequently used. This is not the case for all words.

Similarly, the morphology information is able to provide a primitive solution to another of the differences identified in section 7.2. For five of the assigned words the POS of the correct sense is not given. Therefore, all of the senses from all possible POS must be considered for those words. If the POS is not given, the word form frequency information is particularly helpful. In general, the word forms are

able to provide useful evidence to help to distinguish between the various possible POS. For example, senses which appear in word forms with a *ing* or *ed* suffix are always verb senses. On the training data, by using solely the word form frequency information a 96.6% POS tagging accuracy is achieved on the instances where the POS is not known. This accuracy is slightly lower than can be achieved by a dedicated POS tagger. However, the small accuracy gain, which could have been achieved by incorporating a POS tagger is not considered worthy of the manual resources required, particularly as only five words are not POS tagged and such a development would be of no benefit to the core DURHAM system. Therefore, for the SENSEVAL evaluation, the word form frequencies perform the task of a primitive POS tagger.

The drawback of the word form frequency method is that by making the frequency information more specific, the number of examples in each word form category is less than a single root form category. This results in less accurate frequency information. However, as there are normally no more than four different word form categories, the amount the frequency information is reduced per category is not substantial. Therefore, given the amount of training data available for the SENSEVAL task it is considered beneficial to calculate the frequency information in this way. The merits of applying the method to other disambiguation tasks would depend on the characteristics of each individual task.

7.4 Adaptations to the Contextual Information

One of the most difficult problems for most systems which entered SENSEVAL was the obstacle of adapting to using the HECTOR lexicon. The information available in the dictionary definitions may have been different, or the mappings available between their normal lexicon and HECTOR could cause error. Many other systems may have chosen not to compete because of this difficulty. For DURHAM the process of switching lexicons was much less complex. The only part of the core system dependent on WordNet is the contextual matrix. This section

examines the small adaptations made to the contextual matrix enabling it to be applied to the SENSEVAL task.

The contextual matrix trained on SEMCOR provides the starting point for the SENSEVAL contextual matrix. The adaptation to the new lexicon simply involves replacing the WordNet senses of the 35 words being considered with HECTOR senses. For each HECTOR sense, the semantically closest WordNet sense is identified using the HECTOR / WordNet mappings provided. The HECTOR sense then replaces the WordNet sense in the hierarchy so that the same node in the contextual matrix represents the HECTOR sense. New nodes in the contextual matrix are added in instances where more than one WordNet sense is represented by the same node in the contextual matrix, and where more than one HECTOR sense maps to the same WordNet sense. The newly created node is initially set with all the same contextual scores as the original contextual node. Any WordNet sense which is not mapped onto by a HECTOR sense is removed.

This adaptation of the contextual matrix enables all the HECTOR senses to be considered individually, because all of the senses are represented by a different node in the contextual matrix. The advantage of this method is that the inter-relationships between nodes learnt during training on SEMCOR are able to be applied to SENSEVAL. Therefore, the contextual matrix is already partially trained before any training on the SENSEVAL data commences. Finally, the problem caused by mapping from one lexicon to another is greatly reduced. The contextual matrix trained on SEMCOR merely provides a starting point. Further training of the contextual matrix on the SENSEVAL corpus tunes the matrix for the SENSEVAL task and overcomes any mapping difficulties. This is in contrast with some other systems which choose to identify a sense in the WordNet lexicon and then convert it to a HECTOR sense.

The training of the contextual matrix for SENSEVAL is very similar to the training on SEMCOR. All the words in the sentence are still disambiguated but only the relevant word in the sentence is examined to ascertain whether it is disambiguated correctly. The WordNet *morph* program is used to identify the root

form of all the context words, and the senses from all possible POS are considered.

7.5 An additional Knowledge Source - Clue words

The addition of a third knowledge source for the SENSEVAL task is inspired by one of the evaluation differences identified in section 7.2. Idioms and short phrases are considered a separate sense of the main word in HECTOR, but are considered separate words in WordNet. For example, *dead and buried, bury the hatchet* and *bury one's head in the sand* are all considered as separate senses of *bury* in the HECTOR lexicon. Examining these idioms emphasises a characteristic of WSD which is also true even for lexicons that consider idioms to be a separate word. Whilst in many instances the ambiguity of a word can be very difficult to resolve, there always exist some instances in which the ambiguity can easily be resolved. Some instances are simple to resolve as there exists a word or phrase in the context which provides a strong clue to help resolve the ambiguity. Using the examples above, *dead and, the hatchet* and *one's head in the sand* all provide conclusive clues available in the context, which identify their respective senses.

As a result, the clue words knowledge source is developed to enable accurate ambiguity resolution of these easy instances, and not introduce confusion in the more difficult instances. The way in which the clue words are developed is based on another difference between SENSEVAL and SEMCOR identified in 7.2. The difference is that SENSEVAL is a small scale evaluation in terms of the number of words to be disambiguated. As a result, all clues which help resolve the ambiguity for a word can be manually identified without a huge investment of resources. Also, this task was assigned to a person who was unskilled in any related field. Therefore, the task did not slow down the process of adapting DURHAM to the SENSEVAL task. A discussion of this human resource and its ability to scale up is given in section 7.9.

The manual identification process primarily uses the SENSEVAL training data. However, other textual corpus such as SEMCOR and the Penn Treebank corpus

[Marcus *et al.*, 1993] are used in DURHAM, particularly for words where little or no training data is given. The purpose of the identification process is to identify words in the context which provide evidence for a particular sense or senses of a word. Whichever corpus is being used, its purpose is to act as a trigger to enable the human to identify other similar clues not found in the data. For example, the clue word *troops* can be identified from the corpus for the *take land by force* sense of *seize*. This identification enables the human to identify similar words to *troops* not found in the corpus which could also serve as a clue word, for example *forces, army, military, invasion, marines etc.* All of these words help identify the topic information in which this sense is used and as a result provide useful clues.

In some instances the process of generalising to identify other clues not found in the corpus can be semi-automatic. For example, given the sentence taken from SENSEVAL:

"... brilliant blue sea, custard coloured sand, white yachts and purple mountains in the background."

blue provides a useful clue word to identify the *reflecting a high proportion of light* sense of *brilliant*. However, any colour would provide a similarly useful clue. To prevent the labourious process of entering every possible colour, the WordNet hierarchy is used. In this instance the WordNet node *chromatic* is identified together with a code permitting all hyponyms of that node to be automatically entered as clue words. All colours listed in WordNet are subsumed by *chromatic*. When used in this role, clue words extract similar evidence to the evidence which selectional preferences would be able to produce. The use of clue words in this way provides an alternative approach to selectional preferences. The advantage of the clue words approach is that it is less complex and does not require parsing information.

7.5.1 Position of Clue Words

The developer of a clue words knowledge source must make a compromise between precision and recall. The developer could choose just to add a few clues which

are always correct and have high precision, but low recall. Alternatively, the recall could be increased by adding more clue words, and accepting that in some instances the clue words will provide evidence for the incorrect sense, and thus reducing the precision. In many cases a further piece of evidence can be obtained, which enables the increase in either precision or recall, but not at the expense of the other. This piece of evidence is the position of the clue word relative to the ambiguous word. Specifying the position of the clue word aids in eliminating instances where the clue word appears in the sentence, but is unrelated to the word for which it provides a clue. Thus it enables these words to be included, increasing the recall without causing a decrease to the precision. For example, the following two sentences highlight how the position of *taste* affects its ability to provide a clue for *bitter*.

- "The **taste** of coffee was sweet having come out of the **bitter** weather."
- "The raw lemon left a **bitter taste** in his mouth."

In the first sentence *taste* appears, but is unrelated to *bitter* and therefore does not help resolve the ambiguity. In the second sentence *taste* appears directly after *bitter* and is able to provide a strong clue to resolve the ambiguity.

In some instances, the same clue word can provide evidence for different senses of an ambiguous word by appearing in different positions relative to the ambiguous word. For example, consider the two sentences taken from the SENSEVAL training data:

- "And you'll see four skeletal, toothless old men **shaking hands** and embracing."
- "I know it ain't really polite, but my **hands** are **shaking** so much I'd spill it if I picked it up."

The word *hands* provides a strong clue for the *greeting* sense of *shake* referred to in the first sentence. *Hand* or *hands* appears in 101 out of the 102 training sentences for the *greeting* sense of *shake*. However, *hands* also appears in 22 instances of the

tremble in fear sense of *shake*. The position of the clue word enables a distinction to be drawn relating to the sense that the clue is providing evidence. In most instances *hands* appears after *shake* if it is referring to the *greeting* sense and before *shake* if it is referring to the *tremble in fear* sense.

Other instances occur where the clue word may appear in the same position for more than one sense. Such clues are still beneficial for eliminating some senses even if they are unable to uniquely identify the correct sense. The following example gives two sentences in which *spoon* provides a clue for different senses of *wooden*.

- "Pounding is done with a large **wooden spoon**".
- "The reward for the new captain and his side looked to be worth much more than mere avoidance of the **wooden spoon**".

The example shows that the clue, *spoon*, is unable to distinguish between the *kitchen utensil* and the *coming last in a competition* senses of *wooden*. However, the commonly used *made of wood* sense and the *poor acting* sense can be eliminated from consideration.

Clue words that must appear immediately before or after the ambiguous word are usually referred to as collocates. Collocates have been extensively used by disambiguation systems [Yarowsky, 1995], [Brown *et al.*, 1991], [McRoy, 1992], [Pedersen *et al.*, 1997]. When available, collocates provide extremely strong evidence for a particular sense. However, there are many senses for which collocates are unavailable. For the SENSEVAL task, collocates are manually identified for 35% of all senses.

Many more clue words are identified for the SENSEVAL task which can appear anywhere in the sentence. These clue words aim to capture the more general topic information. Therefore, these clue words are more suited to resolve ambiguous words which possess senses that are specific to different topic domains. In this way, clues can be identified that are likely to appear in the context of a particular topic and unlikely to appear in a different topic. Using the *seize* example considered

earlier, different senses belong to different topic domains. The *take land by force* sense belongs to the military domain, so clues such as *troops, forces, marines, army, over-throw, rule* and *power* also provide clues that this topic domain is being referred to. The *traffic jam* sense belongs to the transport domain, so clues such as *road works, accident, pile up, traffic* and *delays* provide clues for this topic.

Specifying the position of a clue is only one way of aiding the precision/recall compromise. An additional method allows clue phrases as well as single clue words. For example, when *scrap* is followed by the phrase *of difference*, the *no difference* sense of *scrap* is clearly being referred to. However, both of these words could appear after scrap by themselves and not provide such conclusive evidence to aid disambiguation. By incorporating clue phrases, idioms can be identified more clearly. Consider the idiom *"an accident waiting to happen"*. Adding the whole phrase as a clue is better than adding the individual words in the phrase which could occur with a different sense of *accident*. For example, *"The accident waiting room would happen to be near by"*.

7.5.2 Strength of Evidence from Clue Words

Specifying the position of a clue and adding clue phrases has been observed to enable either the precision or recall of clue words to be increased, without a compromise to the other. However, even with these additional pieces of information the compromise between precision and recall still exists. For example, the aforementioned *hands* clue would give mis-leading information in the sentence:

- "You could sense the fear by his shaking hands."

In this sentence the *hands* clue appears after *shaking* but is still referring to the *tremble in fear* sense.

The ideal compromise between precision and recall is dependent on the other knowledge sources also available to aid disambiguation, and the way clue words are incorporated with them. The DURHAM system chooses to make the compromise

in favour of high precision. This compromise is not the best choice for a system in order to achieve as high accuracy as possible in SENSEVAL, but is chosen for several reasons. The sole purpose of clue words within DURHAM is to ensure that the system achieves high accuracy in all the easy instances where the ambiguity can be resolved trivially. Manually identifying these clues is not a time consuming effort. However, continuing to identify further clues that are less accurate is a more time consuming process, and extends beyond the purpose of the clue words knowledge source. A major purpose for competing in SENSEVAL is to evaluate the core part of DURHAM evaluated on SEMCOR. If the clue words knowledge source extends beyond the trivial instances, then the contribution that the core part of DURHAM makes to the overall system is reduced.

Therefore, when available, clue words give very reliable information in assisting to resolve the ambiguity. The most reliable type of clue words are those which take the role of collocates. These must appear immediately before or after the ambiguous word. Clue words which can appear anywhere in the sentence are slightly less reliable and are generally restricted to senses which are domain specific.

These characteristics enable the clue words to complement the remaining two knowledge sources used in the large scale system. As seen in chapter 6, the contextual information is not suited to distinguishing between very fine grained senses. HECTOR is a very fine grained lexicon, and some senses differ only by their syntactic role and not their semantic meaning. Clue words can often make these distinctions. For example, consider the two sentences which refer to different senses of the word *bet.*

- "William Hill stopped taking bets on Thatcher continuing in office."
- "At 7-4, the challenger looks like a good bet."

The first sentence refers to the act of risking money and the second sentence refers to the competitor on which money is risked. It would be difficult for the contextual information to be able to distinguish between these two semantically similar senses. However, *on* frequently follows the first sense of *bet* and *taking*

or *staking* often precedes it. The second sense is often preceded by an adjective such as *good, attractive, best, outside* etc. Therefore, the clue words are able to distinguish between them.

Additionally, the clue words also complement the frequency information by enabling infrequently used senses to be selected. If an infrequent sense possesses a reliable clue word, then that clue provides strong enough evidence to out-weigh the frequency information. This is a useful characteristic as infrequently used senses typically appear infrequently in training data. Therefore, the contextual information, which relies on learning from the training data, is generally unable to provide evidence for an infrequent sense. Using the *wooden* example considered earlier, *spoon* is identified as a useful clue word. Nevertheless, *the made of wood* sense is by far the most frequent, and simply choosing this sense in all training instances achieves 93.9% accuracy on the test data. However, the clue word gives evidence for two of the infrequent senses of *wooden* enabling them to be identified despite the low frequency score. Adding the *spoon* clue word increases the accuracy to 98% on the test data.

7.6 Combining the Clue Words Knowledge Source

The characteristics of the clue words knowledge source have now been discussed. They have been shown to provide strong evidence for a sense when available and complement both the frequency and contextual information knowledge sources. This section examines the way the clue words are combined with the other knowledge sources into the DURHAM system.

Despite the additional knowledge source incorporated for the SENSEVAL task, the general framework for combining knowledge sources remains the same as in the core system. The equations calculating the maximum and minimum score for each sense in the core system were given in equations 5.6 and 5.7. The same equations are used for the SENSEVAL task with the clue words added to the weighted sum.

$$MaxScore_{ij} = A * Frequency_{ij} + B * Max\_CIS(w_{ij}) + C * CW(w_{ij}) \quad (7.1)$$

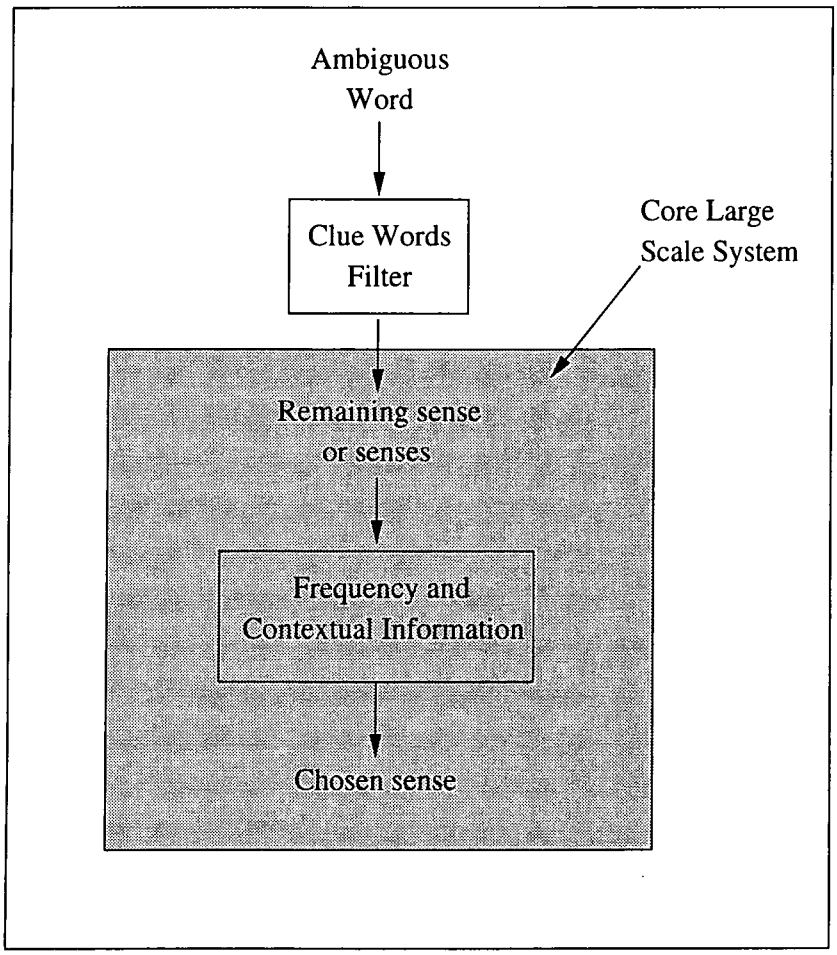
$$MinScore_{ij} = A * Frequency_{ij} + B * Min\_CIS(w_{ij}) + C * CW(w_{ij}) \quad (7.2)$$

where $CW(w_{ij})$ is the clue words score for sense w_{ij}. The clue words score returns zero if no clues are available and one is added to the score for each clue identified in the sentence. This framework for calculating the scores for each sense provides an opportunity for all three knowledge sources to contribute towards the ambiguity resolution. However, tests show that the best way to combine these knowledge sources is to adopt a high value for the constant C used in equations 7.1 and 7.2. In this way if a clue word is available it would have the over-riding effect on the sense choice for that particular ambiguous word. In effect this causes the clue words to act as a filtering system as shown in figure 7.1.

Figure 7.1 shows that the core DURHAM system is used to resolve all senses unable to be resolved by clue words. In general, these are the difficult instances. Considering the clue words knowledge source as a filter changes the approach with which the frequency and contextual information are trained. The analysis of the core system tested on SEMCOR in section 6.5 showed that including the frequency information during the training of the contextual information, assisted in the two knowledge sources complementing each other. The same principle can be applied here, where the evidence from clue words is considered during the training of both the frequency and contextual information. The changes made to the training of both the frequency and contextual information are now discussed

The frequency information is important for the test instances where no clue words have been identified. Therefore, it is beneficial to calculate the frequency information from the subset of training instances where no clue words have been identified. This enables the frequency information to complement the clue words and is favourable to calculating the frequency over the entire training set. If the most frequent sense is identified from the entire training data, then it is less im-

Figure 7.1: Clue words acting as a filter for core large scale system.



portant to find clue words for this most frequent sense. The most frequent sense is likely to be chosen in the instances where no clue words exist. However, the most frequent sense may possess very useful clue words which occur in most of its instances. In this case there is greater benefit in choosing a less frequent sense in an instance where no clue words appear. The following examples helps to explain this idea.

Consider once again the word *shake*. In this example, two different senses will be examined - sense 1: *to shake your head* and sense 2: *to move someone violently*. For the purposes of the example let us suppose that these are the only two possible senses of *shake*. In the training data, 100 instances of sense 1 and 50 instances of sense 2 are identified. The clue word *head* is identified for sense 1 and occurs in 95 out of the 100 training instances. The frequency information is then calculated from the 55 training instances where no clue word can be identified. Out of these 55 instances sense 2 is the most frequent, occurring in 90.9% of all instances. If the frequency information had been calculated from the entire training set, sense 1 would have been the most frequent occurring in 66.7% of all instances. A system which used the single clue word, and frequency information calculated over the entire data set would achieve 66.7% accuracy on the training data set. However, a system which used the single clue word, and frequency information calculated over the subset of training data would achieve 96.7% accuracy on the same data set. This increase in accuracy can be accounted for by an improvement in the way clue words and frequency information complement each other.

Further efforts are also made to ensure that the contextual information complements the other two knowledge sources. However, this is carried out using a different method from the frequency information. Unlike the frequency information, all the training sentences are used to train the contextual information. The reason for this is that other information in the sentence, apart from a clue word, may be present, and this may be able to help train the contextual matrix. Moreover, this approach increases the number of sentences available for training. The drawback of this approach is that the contextual matrix could learn the clue words information. To overcome this problem clue words are omitted from the sentence

during the training of the contextual information. For example, consider the following sentence taken from the SENSEVAL training set for the word *band.*

"For this visit he brings his seven-piece band, including pianist Marcus Roberts, whose album The Truth Is Spoken Here has had considerable success in the States this summer."

NUMBER-piece is identified as a clue word for the sense of band being referred to in this instance. However, there are many other context words which also provide evidence for this sense. Although these other words do not provide strong enough evidence to be identified as clue words, they can help to train the contextual matrix.

7.7 Adaptations to Disambiguation Algorithm

Now that the adaptations to the knowledge sources have been examined, the disambiguation algorithm, which uses the scores from these knowledge source is now considered. The task required to be performed by the disambiguation algorithm is different for the SENSEVAL task compared to the SEMCOR evaluation. This difference stems from the final difference to be considered between the two tasks identified in section 7.2. The SENSEVAL task only requires one word per sentence to be disambiguated, compared to all open class words which must be disambiguated in SEMCOR.

Despite this difference, the disambiguation algorithm is the same for both tasks. All of the context words are still disambiguated according to their WordNet senses. This is carried out as a consequence of the analysis on SEMCOR detailed in section 6.6.2. This analysis shows that choosing the correct sense of the context words enables a large improvement in accuracy. There is no way of training or testing the accuracy for the context words on the SENSEVAL task. However, the SEMCOR results show that DURHAM is able to achieve higher accuracy than any baseline measure. For all the context words, the WordNet *morph* program is used to identify the root form of the word, and senses from all POS are considered.

The drawback of this approach is that it is much less efficient than a system that solely disambiguates one word in each system. In addition, the accuracy gains found on SEMCOR by identifying the correct context may not apply to SENSEVAL. This is because WordNet senses are used for the context words, and HECTOR senses are used for the word being evaluated. This hypothesis is investigated in 7.8.2.

7.8 Results

This section examines the results achieved by DURHAM in SENSEVAL and compares them to the results achieved by some of the other systems which took part in the evaluation. The results presented in this section are limited to those that highlight interesting features of DURHAM. An extensive breakdown of the results is given in [Kilgarrif and Rosenzweig, 2000]. Many evaluation metrics are used to report the results, and these are discussed in section 2.6.2. This section concentrates predominantly on the fine grained results. However, the relative performance of the systems is not changed by using different metrics.

Figure 7.2 shows the performance of all systems that entered the SENSEVAL evaluation. The recall metric refers to the system's accuracy out of all the words in the test set, and the precision metric refers to the accuracy out of all the words that the system attempted. Figure 7.2 shows that DURHAM achieved the highest precision and recall of all the systems entered in SENSEVAL. The system names of the four groups which achieved the highest precision and recall are also shown, this is highlighted in order that these systems can be used for comparison.

A closer examination of the results can be achieved by comparing the four systems identified in figure 7.2 on various subsets of the test data. The precision metric is used to compare different systems as this gives a better indication of the quality of disambiguation. All four systems attempted a high percentage of the test data, so the difference between precision and recall is low. DURHAM attempted all of the test data, so there is no difference between the precision and recall results.

Table 7.2 shows that relative to the other three systems considered, DURHAM

Figure 7.2: The fine grained results for all systems competing in SENSEVAL showing that DURHAM achieved the highest precision and recall.

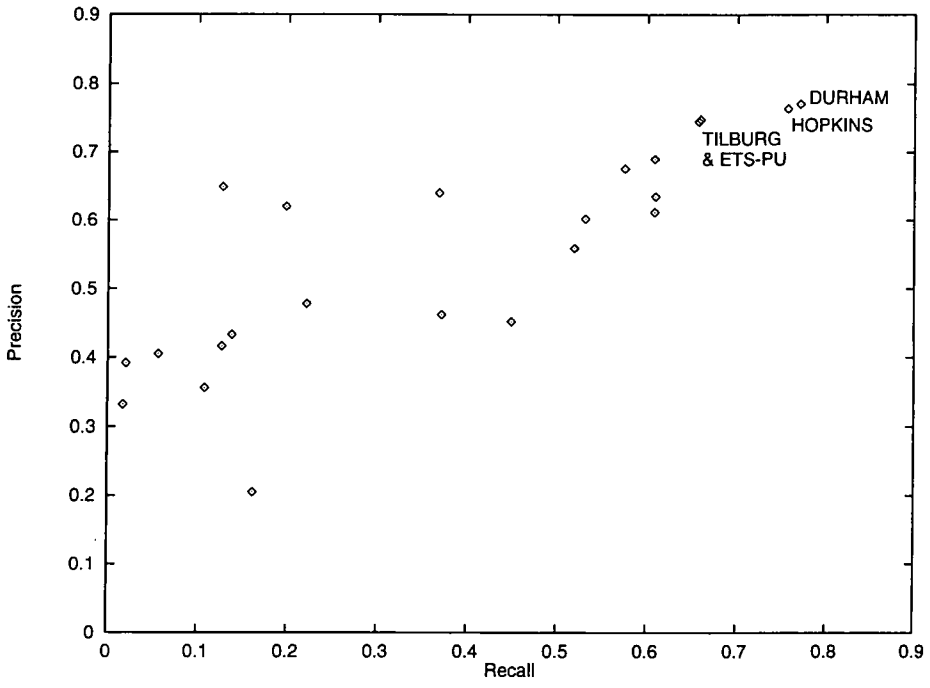


Table 7.2: Comparison of systems on various subsets of the SENSEVAL test data.

System	Nouns	Verbs	Adjectives	Indeterminates	All Words
DURHAM	83.9	70.0	75.2	77.5	77.1
HOPKINS	80.7	71.4	78.4	75.9	76.4
TILBURG	81.9	69.2	72.9	77.1	74.8
ETS - PU	80.7	70.1	72.7	73.5	74.5
FREQ BASELINE	59.9	57.9	64.3	46.7	56.6

performs well on nouns and least well on verbs. HOPKINS performs particularly well on adjectives, with DURHAM 2.3% better than any other system on this subset. Interestingly, DURHAM's relative performance on nouns, verbs and adjectives is the same in both the evaluation on SEMCOR and SENSEVAL - see section 6.7. It is unfortunate that no adverbs were considered in SENSEVAL in order for further evaluation to determine whether this pattern would have continued. Possible reasons for DURHAM achieving a lower accuracy for verbs than nouns are considered in section 6.7. A further reason only applicable to SENSEVAL is concerned with clue words being more difficult to identify for verbs than any other POS.

The indeterminates subset is the group of five words for which no POS is assigned. DURHAM achieves the highest fine grained accuracy on this subset. However, DURHAM assigns a sense with the correct POS in only 95.5% of all instances. Some systems were able to achieve a POS tagging accuracy 3.0% higher than this figure. The lack of a POS tagger as a sense filter in DURHAM is the reason for this. A POS tagger is used in the three systems which achieve higher POS accuracy. This result is to be expected, and demonstrates that although the WSD mechanism performs well as a POS tagger, a dedicated POS tagger is able to perform better.

The frequency baseline accuracy is also included in table 7.2 enabling the accuracy of the systems to be compared with a baseline figure. The frequency baseline figure is used to calculate Kappa for DURHAM.

$$Kappa = \frac{0.771 - 0.566}{1 - 0.566} = 0.472 \tag{7.3}$$

The ITA agreement on SENSEVAL is calculated as 96.5%. Therefore, this figure can be used as an upperbound for an automatic system. This enables the UBAKappa metric, introduced in section 6.1.8, to be calculated.

$$UBAKappa = \frac{0.771 - 0.566}{0.965 - 0.566} = 0.514 \tag{7.4}$$

Both of these measures of Kappa are higher for DURHAM on the SENSEVAL

Table 7.3: The accuracy achieved by the overall DURHAM system and by various components of DURHAM. The figures are expressed as a percentage.

	Description	Fine grained	Onion	Generous	Shake
(1)	Root Form Frequency	56.6	84.6	37.0	23.9
(2)	Word Form Frequency	61.6	85.0	37.0	30.6
(3)	Clue words and word form frequency	73.7	92.5	44.9	71.1
(4)	Contextual scores and word form frequency	69.8	85	50.1	61.8
(5)	Full System	77.1	92.5	50.7	72.5

evaluation than they are for SEMCOR. The reason for this is that the SENSEVAL evaluation considers less ambiguous words and provides more training data for those words. Therefore, higher accuracy is to be expected.

Table 7.3 shows the contribution that each knowledge source makes to the accuracy of the overall system, and for three particular words. Row (2) shows that the overall fine grained accuracy is increased by 5.0% by using word form rather than root form frequencies. This shows that information which aids in disambiguation is available at the morphology level. Row (3) reports the accuracy of a system which uses clue words and the word form frequency information knowledge sources. The word form frequencies have been recalculated to complement clue words as discussed in section 7.6. The substantial increase above the frequency baseline highlights the value of clue words. If the clue words knowledge source was to be considered unacceptable, then row (4) provides interesting results. This shows that DURHAM is able to achieve almost 70% fine grained accuracy without the use of clue words. Such a system would have achieved the fourth highest precision and second highest recall in SENSEVAL. Also the 30% of instances which were incorrectly tagged would include many instances which are considered easy to disambiguate. Nevertheless, it is interesting to note that a system using frequency information and clue words achieves a higher accuracy than a system using frequency information and contextual information. This result would therefore suggest that investing a manual resource into identifying clue words is more beneficial than investing the resource into sense tagging training data required to train

the contextual matrix. Row (5) highlights that the overall system achieves much higher accuracy than any sub-section of it. This suggests that clue words and contextual scores are useful for disambiguating different types of words and so can be successfully combined.

The three individual words presented in table 7.3 are chosen because they highlight interesting characteristics of the system. *Onion* has a high frequency baseline and only 26 training examples are given for that word. This prevents the contextual information from contributing to the accuracy. By contrast, *generous* has a very low frequency baseline and very few clues can be identified to aid disambiguation. These characteristics are ideally suited for the contextual information performing well. This is shown by the 13.1% improvement in accuracy which the contextual information makes. *Shake* is one of the words for which the POS is not given. This is the reason for the large increase in accuracy caused by using word form rather than root form frequency information. *Shake* possesses strong clue words such as *hands* and *head* which help disambiguation and produces 71.1% accuracy. The contextual information is only able to add a further 1.4% to this score in the full system.

7.8.1 SENSEVAL Training data

One of the most pleasing aspects of the results is the high accuracy achieved for the five words for which no training data was given. For these words this system achieves 9.7% higher precision than the next highest system. This result highlights the domain independence of the system. DURHAM has been able to use the training performed on SEMCOR to help disambiguation on SENSEVAL. This suggests that DURHAM could be used for disambiguation using a different lexicon from WordNet whether sense tagged training data was available or not (so long as there are mappings to the WordNet senses.)

For the remaining words in SENSEVAL there exists a large variation in the amount of training data given. For *onion* there are only 26 training sentences, but there are over 1,000 for *accident*. The effect that a large amount of training data

has on the accuracy of this system is now investigated.

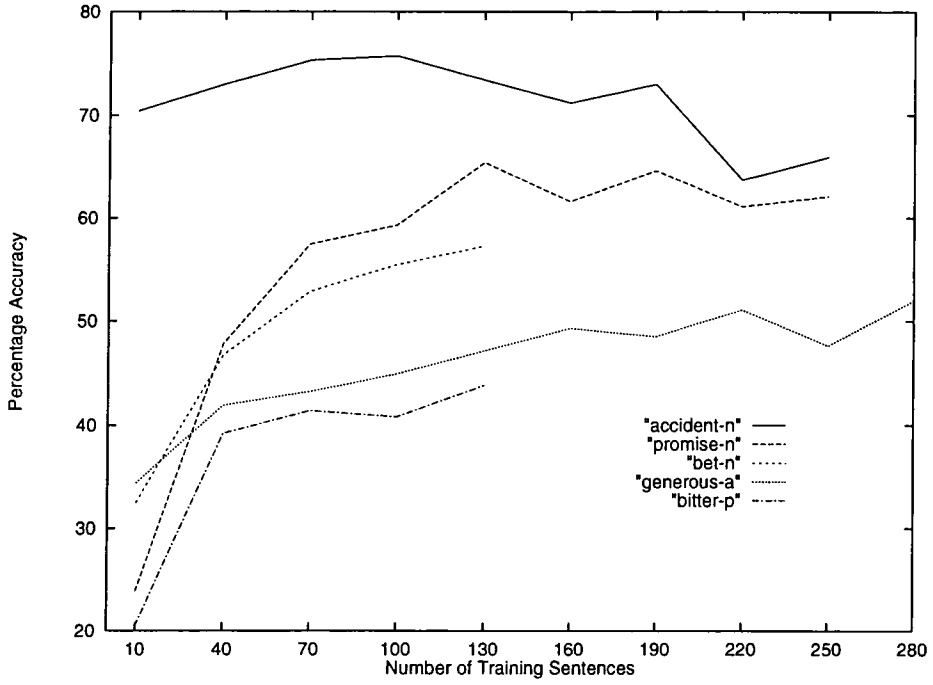
The investigation of the effect that the amount of training data has on accuracy is an important issue for a WSD developer. Additionally, it may be useful for planners of any future disambiguation evaluations providing them with an indication of the effect the quantity of training data has on disambiguation accuracy. This is an important issue, as a vast manual resource is required to produce the training data. Simply correlating the accuracy achieved for each word with the amount of training data supplied, will not enable any conclusion to be drawn about the effect that the quantity of training data has on accuracy. This is due to the existence of many other factors, such as frequency distribution of senses, also affecting accuracy. Instead, the analysis is carried out by choosing a sample of words, and training them using different size subsets of the available training data. In order to maximise the effect of the training data, the clue words knowledge source is not used for this analysis as it is independent of the training data. In addition, the frequency information is calculated using only the subset of training sentences. Figure 7.3 shows the change in accuracy when different amounts of training data are used, for five words in SENSEVAL.

Figure 7.3 shows that for most words there is a sharp increase in accuracy from 10 to 70 training sentences. Generally, accuracy continues to increase at a slower rate up to 130 training sentences. No substantial increase in accuracy is gained by using a higher number of sentences. For *accident* it is apparent that the accuracy actually decreases as more training sentences are used. The contextual information knowledge source has a greater influence on the choice of a sense as more training sentences are used. However, as *accident* has a high frequency baseline, the contextual information is unable to help. The features highlighted in the five words shown on the graph are typical of many other words considered in SENSEVAL.

7.8.2 Correct Context

Section 6.6.2 investigates the importance of choosing the correct sense of the context words for the accuracy on SEMCOR. A similar investigation is now reported on

Figure 7.3: Effect of number of training sentences on accuracy



the SENSEVAL data. Unfortunately, the upper bound measurement used on the SEMCOR analysis can not be calculated for SENSEVAL. During the evaluation, this upper bound was calculated by continually considering the correct sense for the context words. However, the context words are not sense tagged in SENSEVAL so there is no way of knowing which sense is correct. Nevertheless, the investigation is able to measure the importance of correct context relative to a lower bound. As in section 6.6.2 this is performed by continually considering the most frequent sense for the context words.

The system achieves 75.4% fine grained accuracy on SENSEVAL when the most frequent sense of the context words are considered. These frequencies are calculated from WordNet. This accuracy is 1.7% lower than the accuracy achieved by considering all possible senses for the context words. This decrease in accuracy again highlights the importance of the complex disambiguation algorithm employed by DURHAM. However, the difference in accuracy caused by the disambiguation algorithm is less on SENSEVAL than it is on SEMCOR. The reason for this is that the clue words knowledge source is independent of the sense of the contextual words.

Therefore, it makes no difference which sense of the context words is considered if a clue word is able to resolve the ambiguity.

7.9 Are Clue Words a Valid Knowledge Source?

The system which competed in SENSEVAL and the results achieved have now been discussed. This section now moves on to discuss the validity of the system.

Despite DURHAM performing successfully in the SENSEVAL evaluation, the system did receive some criticism. The criticism waged has a sound basis and therefore needs to be considered. The criticism stems from the use of manually identified clue words as a knowledge source. This section presents the case both for and against using clue words in the SENSEVAL evaluation.

7.9.1 Clue Words are not a Valid Knowledge Source

Scalability is the basis on which the manually identified clue words can be criticised. The clue words must be identified individually for each ambiguous word being considered. Therefore, the time required to identify clue words is proportional to the number of ambiguous words being considered. In the WordNet lexicon there are 23,256 ambiguous words. The time required to identify clue words for all of these ambiguous words is too great. Therefore, clue words can not be applied to a large scale system.

The purpose of the SENSEVAL evaluation is to identify systems and approaches which can be applied in NLP systems performing real tasks. Since most NLP tasks are performed on a large scale, it is not beneficial to the evaluation process to consider mechanisms which are unable to scale up. Therefore, clue words should not be included as a knowledge source because it prevents the DURHAM system evaluated in SENSEVAL from scaling up.

7.9.2 Clue Words are a Valid Knowledge Source

Earlier chapters have shown that the core DURHAM system, without the clue words knowledge source, is large scale. This has been highlighted by the evaluation on SEMCOR. In addition a similar system without the clue words knowledge source would achieve fourth highest precision and second highest recall in the SENSEVAL evaluation as shown in section 7.8. However, for all systems using the SENSEVAL training data, including the core DURHAM system without clue words, a scalability problem arises. This is due to the vast quantity of training data provided for 31 of the words considered. The scalability of the training data is a consideration easily over looked. The reason for this is that the manual resource is performed by sense taggers and not by the system developers. The effort required to produce training data on a large scale is far more substantial than that required to produce clue words on a large scale. Human sense taggers report that on average they were able to achieve a speed of 66 instances of a word per hour for the SENSEVAL training data [Krishnamurthy and Nicholls, 2000]. There were on average 426 training instances per word and all instances were sense tagged by two people. Therefore, the manual sense tagging process took approximately two person days per word. The entire process of manually identifying clue words for all the ambiguous words considered in SENSEVAL took less than two days. In addition, the manual sense tagging process must be done by skilled lexicographers. However, no particular skill is required to identify clue words. Not only is the identification of clue words less labour intensive than training data, for DURHAM it is more beneficial. A system which only uses clue words and frequency information achieves 73.7% accuracy. A system using contextual scores learnt on the training data and frequency information achieves 69.8% accuracy.

The SENSEVAL evaluation is therefore not suited to identifying systems which can be applied on a large scale. It is for this reason that there is not a "large scale" category of systems in SENSEVAL. If this category existed, DURHAM would have entered a second system without the clue words knowledge source. SENSEVAL was designed to be a small scale evaluation to encourage participation. This is

necessary as a large scale task is not achievable by many systems.

Within the context of the evaluation, two possible methods exist for identifying the instances where disambiguation is easy. The first is to manually identify the clue words, the second is to automatically identify them using the training data. An automatic approach which requires no sense tagged training data such as that proposed in [Gale *et al.*, 1992c] is not possible in the SENSEVAL evaluation, as a larger discourse would be required. For both feasible methods, the time taken to develop the knowledge source is proportional to the number of words being considered. Therefore, both methods have scalability difficulties. As a result, within the context of a small scale evaluation, the manual identification approach is valid. This approach is chosen for DURHAM as it offers a greater quality of information than could be automatically generated.

7.9.3 A Measure of Scalability

It is important to consider how scalability should be measured so a more definite answer can be determined in the future. In terms of the required human resources, the best indication of scalability which SENSEVAL offers is given by the words where no training data is available. For these words, DURHAM achieves an accuracy 9.7% higher than the next highest system. However, there are only four words in this test set.

In addition, scalability can be measured in terms of coverage. A system capable of being applied to a large scale task must also be able to disambiguate all types of words. DURHAM is one of only four systems which attempts all the test instances given. Nevertheless, there is no doubt that the SENSEVAL evaluation is unable to conclusively determine whether a system is able to scale up.

Although SENSEVAL is able to offer an indication of the scalability of a system, a future evaluation should aim to identify scalable systems more definitely. [Wilks, 2000] proposes that this could be achieved by combining resources to produce a large scale corpus on which WSD systems could be trained and tested. If

the required resources necessary to produce such a corpus were considered "acceptable", it may additionally seem "acceptable" to identify the clue words at the same time. The human sense taggers would identify the clue words as part of the sense tagging process so it would add very little to the cost of resources. The figure of 23,256 ambiguous words in WordNet gives a misleading impression of the size of the task. This is because many of these words are very infrequently used. In SEMCOR only 6241 ambiguous words appear in more than one sense. A large scale clue words resource would be very beneficial to the WSD community as shown by their performance on a small scale. To an extent this process has already taken place in the CIDE project [Harley and Glennon, 1997]. The CIDE dictionary has been developed from a corpus, and does contain clue words in the definition of each sense.

7.10 Conclusion

This chapter has highlighted that there are many differences between the SEMCOR and SENSEVAL evaluation. Despite this, only a few changes are required to enable DURHAM to be evaluated on the SENSEVAL task. Two characteristics of DURHAM have been shown to be applicable on more than one lexicon, and this has facilitated the conversion to a different evaluation task. The contextual matrix trained on SEMCOR has been shown to be beneficial for disambiguating HECTOR senses. This was demonstrated by the high accuracy achieved on the words where no training data was available. Furthermore the training method of the contextual matrix has been shown to be domain independent. This was demonstrated by the ability to use the same mechanism on the HECTOR training data as was used on SEMCOR. These two characteristics demonstrate that the flexibility criterion for success set out in section 2.7 has been fulfilled.

The usability criterion for success refers to the ability of the WSD system to be used by an NLP developer. The required criterion for success is to achieve an accuracy as high as any other system performing the same task. This is partially

fulfilled in chapter 6 which demonstrates that DURHAM is able to achieve high accuracy on a large scale task. However, the SEMCOR evaluation could not be compared with many other systems. This chapter has shown that DURHAM is also able to achieve high accuracy on a small scale task. DURHAM can be compared against many systems in the SENSEVAL evaluation. The accuracy achieved is sufficiently high to fulfil the usability criterion for success.

Chapter 8

Conclusions and Future Work

All the work which was has been undertaken for this piece of research has now been discussed, and the results achieved have been reported. This chapter discusses the conclusions from the work and examines if the criteria for success have been achieved. The chapter then moves on to discuss various directions in which future work could build upon what has been developed.

8.1 Conclusions

The conclusions of this work relates back to the criteria for success outlined in section 2.7. This section investigates these criteria and discusses if the system developed has been able to fulfil these criteria.

Seven specific goals which relate to NLE were identified in section 1.2.2. The criteria for success identified three of these goals as the most relevant for this work. Specific levels of achievement were set for these three goals. The criteria for success also stated that some achievement should be made for the remaining four goals. This section initially discusses the three primary criteria for success for this work, and then also considers the remaining four NLE goals.

8.1.1 Primary Criteria for Success

The three primary criteria for success are each considered individually.

Usability

The accuracy of the WSD system was identified as the feature of primary importance for it to be usable in an NLP system. This criterion stated that the accuracy must be comparable with other WSD systems performing the same task. This work has demonstrated that DURHAM has fulfilled this criterion. On SEMCOR, DURHAM achieved higher accuracy than Agirre and Rigau which is the only system which has performed a comparable large scale task on SEMCOR. Also on SENSEVAL, DURHAM achieved the highest precision and recall on the complete test set. Therefore on both test sets, the usability criterion has been fulfilled.

Flexibility

The WSD system needed to be able to perform in different domains to fulfil the flexibility criterion. Two specific goals were set to measure the flexibility criterion. The first was that a system trained on one lexicon could be applied to a separate lexicon. This achievement has been demonstrated by the contextual matrix trained on SEMCOR, which uses the WordNet lexicon, being applied for SENSEVAL which uses the HECTOR lexicon. The second goal was that the learning algorithm could be applied to more than one lexicon. This was shown by the same learning algorithm being used for SEMCOR and SENSEVAL. Therefore, the flexibility criterion has been achieved.

Scale

The scale criterion determines if the system is able to process all real text. Two specific goals were also set to measure this criterion. The first goal states that all sentences regardless of length can be processed. The second goal states that

all words found in a lexicon can be disambiguated. Both of these specific goals are demonstrated by the evaluation on SEMCOR. The longest sentence on the SEMCOR blind test data contains 158 words. All open class words were disambiguated. Also on SENSEVAL all the test instances for all the words considered were attempted.

8.1.2 Other NLP Goals

The remaining four NLP goals are now considered individually to examine the level of achievement made in these areas.

Robustness DURHAM has achieved a high level of robustness as it is able to fully operate in both the domains within which it was tested.

Maintainability A measure of the systems maintainability was the ease with which it could adapt to a new domain.

Integration DURHAM has not been integrated with other components of a larger system. However, no problems are envisaged with such an integration.

Feasibility DURHAM has been shown to be feasible by operating at an acceptable speed. This was achieved by a complex disambiguation algorithm which performed a directed search through a large search space. The system would not be feasible if all the sense combinations for a sentence were considered individually.

8.2 Future Work

This section examines the various directions in which work in the future could develop further the system reported in this thesis. This is done by firstly considering each of the major components of DURHAM individually, and then by considering the system as a whole.

8.2.1 Frequency Information

The frequency information currently provides a probability for each possible sense given that a particular word has occured in the text. This probabilistic measure is calculated from the training data. The value of this information is dependent on the number of training instances for that particular word. For example, a frequency score of 0.9 for a sense is much more reliable if the sense has occured 90 out of 100 instances rather than 9 out of 10 instances. Future work could examine incorporating the number of training examples into the frequency score. As the frequency score is combined with other non probabilistic measures, the frequency score could also be non probabilistic.

8.2.2 Clue Words

The credibility of manually identified clue words is largely dependent on the question of their scalability. This was discussed in section 7.9. Future work is required to examine the benefit of clue words on a large scale and estimate the investment in man hours required. If this analysis showed clue words to be a credible knowledge source there is scope for further development. Future work could develop a weight associated with a clue word which represented the strength of the evidence the clue provided. In this way the clue words could be incorporated with the other knowledge sources rather than being used as a filter. This would enable a much greater number of clue words to be identified, as the requirement of high precision would be removed.

8.2.3 Contextual Information

Many of the choices made in the development of the contextual information knowledge source were not claimed to be optimal. Therefore, future work could investigate these areas to establish if further improvements could be made. For example, the number of nodes included in the contextual matrix and the choice of those senses could both be further investigated. The mechanism by which the contextual

matrix is trained is similar to neural network learning, and the structure of the contextual matrix most closely resembles a Hopfield network [Krose and van der Smagt, 1993]. Future work could examine how the contextual matrix could be adapted to enable neural networks to provide a large scale knowledge source.

8.2.4 Disambiguation Algorithm

The development of any disambiguation algorithm must consider the compromise between accuracy and efficiency. Future work is likely to move in the direction of higher accuracy at the expense of greater computational requirements. This is because computer hardware improvements reduce the constraints of software efficiency. In this case, the disambiguation algorithm could be developed further. This could be carried out by examining better ways in which to determine the effect that removing a sense will have on the context words. Moreover, future work could examine incorporating a disambiguation algorithm which considers all sense combinations once the number of possibilities has been reduced to below a set level.

8.2.5 Integration

A WSD system is only a component of a larger system. Therefore, an important area for future work is the integration of the system. Integration can be considered at three different levels. Firstly, within WSD other knowledge sources could be integrated such as dictionary definitions and selectional preferences. Also other sub tasks of NLP could be integrated, in particular a POS tagger. This would enable a larger list of possible senses to be accurately considered. Finally, future work could integrate the WSD system into a NLP system performing real tasks. This area of future work has already been planned with some of the techniques developed in this work being incorporated into a NLP system named CONCEPT (formerly known as LOLITA [Morgan *et al.*, 1995]). This demonstrates that the future of NLP is very exciting and may have a significant impact on our everyday lives in the not too distant future.

Appendix A

Training and Test Data

The SEMCOR files used for the training data, validation data and blind test data are now listed.

A.1 Training Data

br-a01	br-c04	br-f03	br-j04	br-j15	br-j54	br-k04	br-k15
br-k26	br-r06	br-a02	br-d01	br-f10	br-j05	br-j16	br-j55
br-k05	br-k16	br-k27	br-r07	br-a11	br-d02	br-f19	br-j06
br-j17	br-j56	br-k06	br-k17	br-k28	br-r08	br-a12	br-d03
br-f43	br-j07	br-j18	br-j57	br-k07	br-k18	br-k29	br-r09
br-a13	br-d04	br-g01	br-j08	br-j19	br-j58	br-k08	br-k19
br-l11	br-a14	br-e01	br-g11	br-j09	br-j20	br-j59	br-k09
br-k20	br-l12	br-a15	br-e02	br-g15	br-j10	br-j22	br-j60
br-k10	br-k21	br-m01	br-b13	br-e04	br-h01	br-j11	br-j23
br-j70	br-k11	br-k22	br-m02	br-b20	br-e21	br-j01	br-j12
br-j37	br-k01	br-k12	br-k23	br-n05	br-c01	br-e24	br-j02
br-j13	br-j52	br-k02	br-k13	br-k24	br-p01	br-c02	br-e29
br-j03	br-j14	br-j53	br-k03	br-k14	br-k25	br-r05	

A.2 Validation Data

br-e22	br-f13	br-f23	br-g18	br-g43	br-h17	br-j34	br-l14
br-n14	br-p24	br-e23	br-f14	br-f24	br-g19	br-g44	br-h18
br-j35	br-l15	br-n15	br-r04	br-e25	br-f15	br-f25	br-g20
br-h09	br-h21	br-j38	br-l16	br-n16	br-e26		

A.3 Blind Test Data

br-f16	br-f33	br-g21	br-h11	br-h24	br-j41	br-l17	br-n17
br-e27	br-f17	br-f44	br-g22	br-h12	br-j29	br-j42	br-l18
br-n20	br-e28	br-f18	br-g12	br-g23	br-h13	br-j30	br-l08
br-n09	br-p07	br-e30	br-f20	br-g14	br-g28	br-h14	br-j31
br-l09	br-n10	br-p09	br-e31	br-f21	br-g16	br-g31	br-h15
br-j32	br-l10	br-n11	br-p10	br-f08	br-f22	br-g17	br-g39
br-h16	br-j33	br-l13	br-n12	br-p12			

Glossary

CIS Contextual information score

Contextual Information The novel knowledge source introduced in this work.

Contextual Score The score between two nodes in the WordNet hierarchy

IE Information extraction

IR Information retrieval

ITA Inter tagger agreement

MT Machine Translation

NIE No Intersection Elimination

NMSE Normalised Max Score Elimination

NLE Natural Language Engineering

NLP Natural Language Processing

POS Part of speech

UBAAKappa Upper bound adjusted Kappa

WSD Word sense disambiguation

Bibliography

[Agirre and Rigau, 1995] E. Agirre and G. Rigau, "A Proposal for Word Sense Disambiguation using Conceptual Distance", *1st Intl. Conf. on recent Advances in NLP*, 1995.

[Agirre and Rigau, 1996] E. Agirre and G. Rigau, "Word Sense Disambiguation Using Conceptual Density", in *The proceedings of COLING 1996*, pages 16–22, Copenhagen, Denmark, 1996.

[Atkins, 1993] S. Atkins, "Tools for computer-aided lexicography: the Hector project", in *Computational Lexicography: COMPLEX '93*, Budapest, 1993.

[Back and Schwefel, 1993] T. Back and H.-P. Schwefel, "An overview of Evolutionary Algorithms for Parameter Optimization", *Evolutionary Computation*, 1(1):1–23, 1993.

[Bar-Hill, 1960] Y. Bar-Hill, "Automatic translation of languages", in *Advances in computers*, Academic Press, New York, 1960.

[Basili *et al.*, 1992] R. Basili, M. Pazienza, and P. Velardi, "A shallow syntactic analyser to extract word associations from corpora", *Literacy and Linguistic Computing*, 7(2):114–124, 1992.

[Beardon, 1989] C. Beardon, *Artificial intelligence terminology: A reference guide*, Ellis Horwood, 1989.

[Boguraev *et al.*, 1995] B. Boguraev, R. Garigliano, and J. Tait, "Editorial", *Journal of natural language engineering*, 1(1), 1995.

[Brill, 1992] E. Brill, "A simple rule-based part-of-speech tagger", in *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy, 1992.

[Brill, 1995] E. Brill, "Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging", *Computational Linguistics*, 21(4):543–566, 1995.

[Brown *et al.*, 1988] P. Brown, J.Cocke, S. D. Pietra, V. D. Pietra, F. Jelinek, R. Mercer, and P. Roosin, "A statistical approach to language translation", in *Proceedings of the International Congress on Computational Linguistics*, pages 71–75, Budapest, Hungary, 1988.

[Brown *et al.*, 1991] P. F. Brown, S. A. Della Pietra, and V. J. Della Pietra, "Word Sense Disambiguation using Statistical Methods", in *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 264–270, 1991.

[Bruce and Wiebe, 1994] R. Bruce and J. Wiebe, "Word sense disambiguation using decomposable models", in *Proceedings of 32nd Annual Meeting of the Assoiciation for Computation Linguistics*, pages 139–146, Las Cruces, New Mexico, 1994.

[Bruce and Wiebe, 1998] R. Bruce and J. Wiebe, "Word sense distinguishability and inter-coder agreement", in *Proceedings on 3rd Conference on Empirical Methods in Natural Language Processing (EMNLP-98). Association for Computational Linguistics SIGDAT*, Granada, Spain, June 1998.

[Buitelaar, 1998] P. Buitelaar, *CoreLex: Systematic polysemy and underspecification*, PhD thesis, Brandeis University, Boston, 1998.

[Callaghan, 1998] P. Callaghan, *An evaluation of LOLITA and related natural language processing systems*, PhD thesis, Durham University, 1998.

[Calmet and Campbell, 1993] J. Calmet and J. A. Campbell, "Artificial intelligence and symbolic mathematical computations", in *Proceedings of the conference on artificial intelligence and symbolic mathematical computations, volume 737 of Lecture Notes in Computer Science*, pages 1–19, Springer-Verlag, 1993.

[Chang *et al.*, 1992] J.-S. Chang, Y.-F. Luo, and K.-Y. Su, "GPSM: A Generalised Probabilistic Semantic Model For Ambiguity Resolution", in *Proceedings of ACL-92*, pages 177–184, University of Delaware, USA, June 1992.

[Chapman, 1977] R. Chapman, *Roget's International thesaurus*, Harper and Row, New York, fourth edition, 1977.

[Charniak, 1994] E. Charniak, *Statistical Language Learning*, MIT Press, 1994.

[Chodorow and Byrd, 1985] M. S. Chodorow and R. J. Byrd, "Extracting Semantic Hierarchies from a large On-Line Dictionary", in *Proceedings of the Association for Computational Linguistics*, volume 23, pages 299–304, 1985.

[Chomsky, 1965] N. Chomsky, *Aspects of the theory of syntax*, MIT Press, Cambridge, MA, 1965.

[Cowie *et al.*, 1992] J. Cowie, J. Guthrie, and L. Guthrie, "Lexical Disambiguation Using Simulated Annealing", in *Proceedings of COLING-92*, pages 359–365, 1992.

[Crane, 1987] S. Crane, *The red badge of courage*, Chelsea House Publishers, New York, 1987.

[Cunningham *et al.*, 1996] H. Cunningham, R. Gaizauskas, and Y. Wilks, "GATE - a general architecture for text engineering", in *Proceedings of the 16th Conference on Computational Linguistics (COLING-96)*, Copenhagen, 1996.

[Cunningham *et al.*, 1998] H. Cunningham, M. Stevenson, and Y. Wilks, "Implementing a Sense Tagger within a General Architecture for Text Engineering", in *Proceedings of the Third Conference on New Methods in Language Engineering (NeMLaP-3)*, pages 59–72, Sydney, Australia, 1998.

[EC, 1991] EC, "Linguistic research and engineering in the framework programme 1990-1994: Technical background document", Technical report, European Community, 1991, Report MCCS-87-96.

[Ellman *et al.*, 2000] J. Ellman, I. Klincke, and J. Tait, "Word sense disambiguation by information filtering and extraction", *Computers and the Humanities - Special Issue on Senseval*, 34, 2000.

[Fellbaum *et al.*, 1996] C. Fellbaum, J. Grabowski, S. Landes, and A. Baumann, "Matching words to senses in WordNet: Naive vs expert differentiation of senses", in C. Fellbaum, editor, *WordNet: An electronic lexical database and some of its applications*, MIT Press, Cambridge, MA, 1996.

[Fellbaum, 1996] C. Fellbaum, editor, *WordNet: An electronic lexical database and some of its applications*, MIT Press, Cambridge, MA, 1996.

[Fellbaum, 1997] C. Fellbaum, *WordNet: An electronic lexical database*, MIT Press, Cambridge, MA, 1997.

[Fujii *et al.*, 1988] A. Fujii, K. Inui, T. Tokunaga, and H. Tanaka, "Selective sampling for example-based word sense disambiguation", *Computational Linguistics*, 24(4):573–598, 1988.

[Fujii *et al.*, 1996] A. Fujii, K. Inui, T. Tokunaga, and H. Tanaka, "Selective Sampling of Effective Example Sentence Sets for Word Sense Disambiguation", in *Proceedings of the Fourth Workshop on Very Large Corpora WVLC-4*, pages 56–69, 1996.

[Fujii, 1998] A. Fujii, *Corpus-based word sense disambiguation*, PhD thesis, Tokyo Institute of Technology, 1998.

[Gale and Church, 1991a] W. Gale and K. Church, "Identifying word correspondences in parallel texts", in *Proceedings of the DARPA Conference on Speech and Natural Language*, 1991.

[Gale and Church, 1991b] W. Gale and K. Church, "A program for aligning sentences in bilingual corpora", in *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 177–184, 1991.

[Gale *et al.*, 1992a] Gale, Church, and Yarowsky, "A Method for Disambiguating Word Senses in a Large Corpus.", *Computers and the Humanities.*, 26:415–439, 1992.

[Gale *et al.*, 1992b] W. Gale, K. Church, and D. Yarowsky, "Estimating upper and lower bounds on the performance of word sense disambiguation programs", in *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL- 92)*, pages 249–256, Newark, DE, 1992.

[Gale *et al.*, 1992c] W. Gale, K. Church, and D. Yarowsky, "One sense per discourse", in *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 233–237, Harriman, NY, 1992.

[Gale *et al.*, 1995] Gale, Church, and Yarowsky, "Discrimination Decisions for 100,000 dimensional spaces", *Annals of Operations Research*, 55:323–344, 1995.

[G.Cottrell, 1984] G.Cottrell, "A model of lexical access of ambiguous words", in *In proceedings of the national conference on artificial intelligence (AAAI-84)*, pages 61–67, 1984.

[Gomez, 1997] F. Gomez, "Linking WordNet Verb Classes to Semantic Interpretation", Technical report, Dept Computer Science University of Central Florida, 1997, Report UCF-CS-TF-97-03.

[Grefenstette, 1997] G. Grefenstette, "Short query linguistic expansion techniques: Palliating one-word queries by providing intermediate structure to text", in *Information extraction - Lecture Notes in Artificial Intelligence - 1299*, Springer-Verlang, 1997.

[Hanks, 1979] P. Hanks, *Collins English Dictionary*, Collins, London and Glasgow, 1979.

[Hanks, 1993] P. Hanks, "Lexicography: Theory and practice", *Dictionaries*, 14:97–112, 1993.

[Hanks, 2000] P. Hanks, "Do word meanings exist", *Computers and the Humanities - Special Issue on Senseval*, 34, 2000.

[Harley and Glennon, 1997] A. Harley and D. Glennon, "Sense tagging action in action: Combining different tests with additive weightings", in *Proceedings of SIGLEX workshop "Tagging text with lexical semantics"*, pages 74–78, Washington D.C., April 1997.

[Hawkins, 2000] P. Hawkins, "Large Scale WSD using Learning Applied to Senseval", *Computers and the Humanities - Special Issue on Senseval*, 34, 2000.

[Hearst, 1991] M. Hearst, "Noun Homograph Disambiguation using Local Context in Large Text Corpora", *Using Corpora*, pages 1–23, 1991.

[Hirst, 1994] G. Hirst, "Jumping to conclusions: Psychological reality and unreality in a word disambiguation program", in *Proceedings of the 6th Annual Conference of the Cognitive Science Society*, pages 179–182, 1994.

[Hornby, 1963] A. Hornby, *The advanced learner's dictionary of English*, Oxford University Press, Oxford, 1963.

[Hutchins and Somers, 1992] J. Hutchins and H. Somers, *Introduction to Machine Translation*, Academic Press, 1992.

[Hwee Tou Ng and Hian Beng Lee, 1996] Hwee Tou Ng and Hian Beng Lee, "Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach", in *Proceedings of the Association of Computational Linguistics*, pages 40–47, Santa Cruze, CA, 1996.

[Ide and Veronis, 1998] N. Ide and J. Veronis, "Automatic word sense discrimination", *Computational Linguistics*, 24(1):97–125, March 1998.

[k Kawamoto and Anderson, 1994] A. k Kawamoto and J. A. Anderson, "Lexical Access Using a Neural Network", in *Proceedings of the 6th Annual Conference of the Cognitive Science Society*, pages 204–213, 1994.

[Karov and Edelman, 1996] Y. Karov and S. Edelman, "Similarity-based Word Sense Disambiguation", *Computational Linguistics*, 24(1):41–59, 1996.

[Katz and Fodor, 1964] J. Katz and J. Fodor, "The structure of a semantic theory", in J. Katz and J. Fodor, editors, *The structure of language*, Prentice Hall, New York, 1964.

[Kaufmann, 1995] M. Kaufmann, editor, *Proceedings of the sixth Message Understanding Conference (MUC-6)*, San Mateo, CA, 1995.

[Key-yih Su *et al.*, 1992] Key-yih Su, Jing-Shin Chang, and Yi-Chung Lin, "A Discriminative Approach for Ambiguity Resolution Based on a Semantic Score Function", in *International Conference of Spoken Language Processing*, 1992.

[Kilgarrif and Rosenzweig, 2000] A. Kilgarrif and J. Rosenzweig, "Framework and results for English", *Computers and the Humanities - Special Issue on Senseval*, 34, 2000.

[Kilgarriff, 1992] A. Kilgarriff, *Polysemy*, PhD thesis, University of Sussex, 1992.

[Kilgarriff, 1993] A. Kilgarriff, "Dictionary word sense distinctions: An enquiry into their nature", *Computers and the Humanities*, 26:356–387, 1993.

[Kilgarriff, 1997a] A. Kilgarriff, "Foreground and Background Lexicons and Word Sense Disambiguation for Information Extraction", in *International Workshop on Lexically Driven Information Extraction.*, pages 51 – 62, Frascati, Italy, July 1997.

[Kilgarriff, 1997b] A. Kilgarriff, "What is Word Sense Disambiguation Good For?", in *Natural Language Processing Pacific Rim Sym-*

posium, pages 209 – 214, Phuket, Thailand, December 1997.

[Kilgarriff, 1998a] A. Kilgarriff, "Gold Standard Datasets for Evaluating Word Sense Disambiguation Programs", in *Computer Speech and Language, Special issue on evaluation*, 1998.

[Kilgarriff, 1998b] A. Kilgarriff, "SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs", in *Proceedings of LREC*, pages 581 – 588, Granada, Spain, May 1998.

[Krishnamurthy and Nicholls, 2000] R. Krishnamurthy and D. Nicholls, "Peeling an onion: the lexicographer's experience of manual sense-tagging", *Computers and the Humanities - Special Issue on Senseval*, 34, 2000.

[Krose and van der Smagt, 1993] B. Krose and P. van der Smagt, *An Introduction to Neural Networks*, University of Amsterdam, fifth edition, 1993.

[Krovets and Croft, 1992] R. Krovets and W. Croft, "Lexical ambiguity and information retrieval", *ACM Transaction on information systems*, 10(2):115–141, 1992.

[Krovetz, 2000] R. Krovetz, "More that one sense per discourse", *Computers and the Humanities - Special Issue on Senseval*, 34, 2000.

[Landes *et al.*, 1996] S. Landes, C. Leacock, and R. Tengi, "Building a semantic concordance of English", in C. Fellbaum, editor, *WordNet: An electronic lexical database and some of its applications*, MIT Press, Cambridge, MA, 1996.

[Leacock and Chodorow, 1998] C. Leacock and M. Chodorow, *WordNet - An electronic lexical database*, chapter Combining local con-

text and WordNet similarity for word sense identification, pages 265–283, MIT Press, 1998.

[Lesk, 1986] M. E. Lesk, "Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from a Ice Cream Cone", in *Proceedings of the ACM SIGDOC Conference, Toronto, Ontario*, pages 24–26, 1986.

[Locke and Booth, 1955] W. Locke and D. Booth, editors, *Machine translation and languages*, John Wiley & Sons, 1955.

[Lyon, 1994] C. Lyon, *The Representation of Natural Language to Enable Neural Networks to Detect Syntactic Features*, PhD thesis, Hertfordshire University, 1994.

[Lytinen, 1986] S. Lytinen, "Dynamically combining syntax and semantics in natural language processing", in *Proceedings of AAAI-86*, pages 574–578, 1986.

[Marcus *et al.*, 1993] M. Marcus, B. Santorini, and M. Marcinkiewicz, "Building a large annotated corpus of English: the Penn treebank", *Computational Linguistics*, 19, 1993.

[McRoy, 1992] S. W. McRoy, "Using Multiple Knowledge Sources for Word Sense Disambiguation", in *Computational Linguistics*, volume 18, pages 1–31, March 1992.

[Melamed and Resnik, 2000] D. Melamed and P. Resnik, "A proposal for SENSEVAL scoring scheme", *Computers and the Humanities - Special Issue on Senseval*, 34, 2000.

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

[Miller and Teibel, 1991] G. Miller and D. Teibel, "A Proposal for Lexical Disambiguation", in *Proceedings of DARPA Speech and Language Workshop*, pages 395 – 399, 1991.

[Morgan *et al.*, 1995] R. Morgan, R. Garigliano, P. Callaghan, S. Poria, M. Smith, A. Urbanowicz, R. Collingham, M. Costantino, C. Cooper, and the LOLITA Group, "Description of the LOLITA System as used in MUC-6", in *Proceedings of the sixth Message Understanding Conference (MUC-6)*, pages 71–87, November 1995.

[Newell and Simon, 1976] A. Newell and H. A. Simon, "Computer science as empirical inquiry: symbols and search", *Communications of the ACM*, 19(3):113–126, 1976.

[Paliouras *et al.*, 1999] G. Paliouras, V. Karkaletsis, and C. D. Sptropoulos, "Learning rules for large vocabulary word sense disambiguation", in *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI*, pages 674–679, volume 2, Stockholm, Sweden, 1999.

[Pedersen *et al.*, 1997] T. Pedersen, R. Bruce, and J. Wiebe, "Sequential Model Selection for Word Sense Disambiguation", in *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP-97)*, Washington DC, April 1997.

[Poria, 1999] S. Poria, *An Engineering Approach to Knowledge Acquisition by the Interactive Analysis of Dictionary Definitions*, PhD thesis, Durham University, 1999.

[Procter, 1978] P. Procter, editor, *Longman dictionary of contemporary english*, Longman Group, London, 1978.

[Procter, 1995] P. Procter, editor, *Cambridge international dictionary of english*, Cambridge University Press, Cambridge, 1995.

[Resnik and Yarowsky, 1997] P. Resnik and D. Yarowsky, "A perspective on word sense disambiguation methods and their evaluation", in *In Proceedings of ACL SIGLEX '97 Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, pages 79–86, Washington DC, America, 1997.

[Resnik, 1995a] P. Resnik, "Disambiguating Noun Groupings with Respect to Wordnet Senses", in *Third Workshop on Very Large Corpora*, June 1995.

[Resnik, 1995b] P. Resnik, "Using Information Context to Evaluate Semantic Simularity in a Taxonomy", in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[Richardson *et al.*, 1994] R. Richardson, A. Smeaton, and J. Murphy, "Using WordNet as a knowledge base for measuring semantic similarity between words", Technical report, Dublin City University, School of Computer Applications, Dublin, Ireland, 1994, Working Paper CA-1294 ftp://ftp.compapp.dcu.ie/pub/w-papers/1994/CA1294.ps.Z.

[Rigau and Agirre, 1995] G. Rigau and E. Agirre, "Disambiguating Bilingual Nominal Entries against WordNet", *Workshop on the Computational Lexicon - ESSLLI 95*, 1995, Available as `http://xxx.lanl.gov/abs/cmp-lg/9510004`.

[Rumelhart *et al.*, 1986] D. E. Rumelhart, G. E. Hinton, and J. L. McClelland, *Parallel Distributed Processing*, MIT Press, 1986.

[Sampson, 1987] G. Sampson, "Probabilistic model of analysis", in R. Garside, G. Leech, and G. Sampson, editors, *The computational analysis of English: A corpus-based approach*, chapter 2, pages 16–29, Longman, 1987.

[Sanderson, 1994] M. Sanderson, "Word sense disambiguation and information retrieval", in *Proceedings of the 17th ACM SIGIR Conference*, pages 142–151, Dublin, Ireland, 1994.

[Schütze and Pedersen, 1995] H. Schütze and J. Pedersen, "Information retrieval based on word senses", in *Proceedings of SDAIR'95*, Las Vegas, 1995.

[Siegel, 1956] S. Siegel, *Nonparametric statistics for the behavioral sciences*, McGraw-Hill Book Company, 1956.

[Slator and Wilks, 1987] B. M. Slator and Y. Wilks, "Towards semantic structures from dictionary entries", Technical report, Computing Research Laboratory, New Mexico State University, 1987, Report MCCS-87-96.

[Stevenson *et al.*, 1998] M. Stevenson, H. Cunningham, and Y. Wilks, "Sense tagging and language engineering", in *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 185–189, Brighton, UK, 1998.

[Stevenson, 1999] M. Stevenson, *Multiple knowledge sources for word sense disambiguation*, PhD thesis, University of Sheffield, 1999.

[Sussna, 1993] M. Sussna, "Word Sense Disambiguation for Free-Text Indexing Using a Massive Semantic Network", in *Proceedings of the 2nd International Conference on Information and knowledge management*, pages 67–74, 1993.

[Véronis and Ide, 1990] J. Véronis and N. Ide, "Word sense disambiguation with very large neural networks extracted from machine readable dictionaries", in *Proceedings of the 13th International Conference on Computational Linguistics, COLING'90*, volume 2, pages 389–394, Helsinki, 1990.

[Véronis and Ide, 1995] J. Véronis and N. Ide, "Large Neural Networks for the resolution of lexical ambiguity", in P. Saint-Dizier and E. Viegas, editors, *Computational Lexical Semantics. Natural Language Processing Series*, pages 251–269, Cambridge University Press, 1995.

[Voorhees, 1993] E. M. Voorhees, "Using WordNet to Disambiguate Word Senses for Text Retrieval", in *Proceedings of the 16th Annual International ACM SIGIR Conference - Research and Development in Information Retrieval*, pages 171–180, 1993.

[Voorhees, 1998] E. Voorhees, *WordNet - An electronic lexical database*, chapter Using WordNet for Text Retrieval, pages 285–303, MIT Press, 1998.

[Weaver, 1955] W. Weaver, "Machine translation and languages", in Locke and Booth [1955], pages 15–23.

[Wiebe *et al.*, 1997] J. Wiebe, J. Maples, L. Duan, and R. Bruce, "Experience in WordNet sense tagging in the Wall Street Journal", in *Proceedings of ANLP-97 Workshop, Tagging Text with Lexical Semantics: Why, What, and How? Association for Computational Linguistics SIGLEX, Washington, D.C.*, pages 8–11, April 1997.

[Wilks and Stevenson, 1996] Y. Wilks and M. Stevenson, "The Grammar of Sense: Is word-sense tagging much more than part-of-speech tagging?", Technical Report CS-96-05, University of Sheffield, 1996, Available as `http://xxx.lanl.gov/ps/cmp-lg/9607028`.

[Wilks and Stevenson, 1997a] Y. Wilks and M. Stevenson, "Combining Independent Knowledge Sources for Word Sense Disambiguation", in *Proceddings of the Conference Recent Ad-*

vances in Natural Language Processing, pages 1–7, Tzigov Chark, Bulgaria, 1997.

[Wilks and Stevenson, 1997b] Y. Wilks and M. Stevenson, "Sense Tagging: Semantic Tagging with a Lexicon", in *Proceedings of the SIGLEX Workshop "Tagging Text with Lexical Semantics: What, why and how?"*, pages 47–51, Washington, D.C., April 1997, Available as `http://xxx.lanl.gov/ps/cmp-lg/9705016`.

[Wilks and Stevenson, 1998] Y. Wilks and M. Stevenson, "Word sense disambiguation using optimised combinations of knowledge sources", in *The 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-98)*, pages 1398–1402, Montreal, Canada, 1998.

[Wilks and Stevenson, 1999] Y. Wilks and M. Stevenson, "Combining weak knowledge sources for sense disambiguation", in *Proceedings of IJCAI-99*, 1999.

[Wilks, 1968] Y. Wilks, *Argument and proof*, PhD thesis, Cambridge University, 1968.

[Wilks, 1978] Y. Wilks, "Making preferences more active", *Artificial Intelligence*, 11(3):197–223, 1978.

[Wilks, 2000] Y. Wilks, "Is word sense disambiguation just one more NLP task?", *Computers and the Humanities - Special Issue on Senseval*, 34, 2000.

[Winston, 1992] P. Winston, *Artificial Intelligence*, Addison-Wesley, third edition, 1992.

[Yarowsky, 1992] D. Yarowsky, "Word-Sense disambiguation using statistical models of roget's categories trained on large corpora",

in *In Proceedings of COLING-92.*, pages 454–460, Nantes, 1992.

[Yarowsky, 1994] D. Yarowsky, "Decision Lists for Lexical Ambiguity Resolution:Application to Accent Restoration in Spanish and French", in *In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics.*, pages 88–95, 1994.

[Yarowsky, 1995] D. Yarowsky, "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods", in *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[Yarowsky, 1996] D. Yarowsky, *Three algorithms for lexical ambiguity resolution*, PhD thesis, School of Computer and Information Science, University of Pennsylvania, 1996.

[Yngve, 1955] V. Yngve, "Machine translation and languages", in Locke and Booth [1955], pages 208–226.

[Zernik and Jacobs, 1990] U. Zernik and P. Jacobs, "Tagging for learning: Collecting semantic relations from a corpus", in *Proceedings of the 13th International Conference on Computational linguistics (COLING - 90)*, pages 34–37, Helsinki, Finland, 1990.