2004

A Similarity Based Concordance Approach to Word Sense Disambiguation

Ramakrishnan B. Guru
Eastern Illinois University

THESIS REPRODUCTION CERTIFICATE

TO: Graduate Degree Candidates (who have written formal theses)

SUBJECT: Permission to Reproduce Theses

The University Library is receiving a number of request from other institutions asking permission to reproduce dissertations for inclusion in their library holdings. Although no copyright laws are involved, we feel that professional courtesy demands that permission be obtained from the author before we allow these to be copied.

PLEASE SIGN ONE OF THE FOLLOWING STATEMENTS:

Booth Library of Eastern Illinois University has my permission to lend my thesis to a reputable college or university for the purpose of copying it for inclusion in that institution's library or research holdings.

09/06/07.

Date

I respectfully request Booth Library of Eastern Illinois University **NOT** allow my thesis to be reproduced because:

_____

_____

_____

_____ _____

Author's Signature Date

This form must be submitted in duplicate.

A Similarity Based Concordance Approach To Word Sense Disambiguation

(TITLE)

BY

Ramakrishnan B Guru

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

Master of Science in Technology

IN THE GRADUATE SCHOOL, EASTERN ILLINOIS UNIVERSITY
CHARLESTON, ILLINOIS

2004

YEAR

I HEREBY RECOMMEND THAT THIS THESIS BE ACCEPTED AS FULFILLING
THIS PART OF THE GRADUATE DEGREE CITED ABOVE

September 7, 2004
DATE

9-9-04
DATE

Thesis Panel members

Dr. Samuel Guccione
(Thesis Advisor)

Dr. Peter Ping Liu

Dr. Peter Andrews

THESIS

A Similarity Based Concordance Approach to Word Sense Disambiguation

Ramakrishnan B Guru

School of Technology

Eastern Illinois University

Abstract

This study attempts to solve the problem of Word Sense Disambiguation using a combination of statistical, probabilistic and word matching algorithms. These algorithms consider that words and sentences have some hidden similarities and that the polysemous words in any context should be assigned to a sense after each execution of the algorithm. The algorithm was tested with sufficient sample data and the efficiency of the disambiguation performance has proven to increase significantly after the inclusion of the concordance methodology.

Acknowledgements

I would like to express my sincere gratitude and thanks to Dr. Samuel Guccione for his extensive guidance, support and encouragement throughout this research, without whom a success like this, I would have never seen.

I would like to thank Dr. Peter Ping Liu, my graduate co-ordinator as well as a member of the review panel for this thesis for his constant guidance during the course of my Masters program. I would also like express my heartfelt thanks to Dr. Peter Andrews, a plethora of information in the field of Mathematics and Computer Science, for being in the review panel.

I would also like to extend a special thanks to my friend Jeslina J Raj who is pursuing her Doctoral Program in Psychology, first of all for being a great friend and also for being a huge source of encouragement during every phase of my thesis and also to my roommate Prashanth Padakanti who is also pursuing his Doctoral degree in Chemistry for all the interesting discussions we had during the course of this research.

Dedication

In strong dedication to my dad Shri. R. B. Tilak and my mom Shri. Prameela for their love, affection, everlasting support & guidance throughout my career and so to my ever loved Grandfather Late. Shri. T.S. Ramachandran Iyer, who is not physically with us today but will be in our hearts forever.

Table of Contents

List of Tables

List of Figures

Introduction

The task of natural language processing has reached unforeseen successes in the recent years. The earliest computers were number processors and one could substitute them with programmable calculators since either of them had a similar Input-Process-Output (I-P-O) cycle and had similar applications and resources to work upon. The expected output from a computer was also not as challengeable as it is today because researchers weren't sure about what a computer was capable of. Research areas like Natural Language Processing (NLP) which are developing concepts of today were a distant dream then. Since the inception of the computer, till today, most computers have represented the linguistic aspects of computing in a non-linguistic way. So, most computers that were put into use for natural language research were just counting machines. They could count word occurrences and similarities in patterns from much more text than a human brain could process at a given time and they never ceased because they were never tired. Some of the earliest works that came to be known as computational linguistics did exactly this kind of counting.

Early researchers used computers to compile statistics about texts and also to trace occurrences. Slowly the research in NLP started branching into more subsets and now, making a computer understand word senses from a text, what we call the Word Sense Disambiguation, has been a significant area of research. This requires the agent(explained in detail later in this section) to identify the data from the input set - sentence by sentence or word by word and then follow an algorithm to do the required action which is called disambiguation.

The research described herein is to design an efficient disambiguation technique for multiple senses of a word. This text extends from discussion on some early approaches to disambiguation to the recent advances and also proposes a unique concordance approach towards solving the problem of word sense disambiguation.

WSD Applications

Machine Translation

Machine Translation (MT) refers to the process of translating text or tagged corpora from one language to another without any alterations to the meaning. In fact, the implementation of MT is not as easy as its definition sounds. A typical MT application has three attributes: two monolingual corpora and a bilingual dictionary. The bilingual dictionary maps the two monolingual corpora with words from each corpus and their appropriate translation in the other language. This is not an easy job because most of the languages differ in their form, nature and usage. The concept of words with multiple meanings makes the problem worse. These words cause havoc in a machine translating environment. With the varied number of languages existing in the world, it becomes very difficult for any translator to translate from one form to another. Moreover, the form and sentence formation differ widely among languages. So, it becomes a very difficult task to express the sentence in a particular language and in the same manner and sense in another language. Word

sense disambiguation systems help the MT system in disambiguating words from one language to another and also within the same language. This helps MT a great deal because the heart of MT lies in translating correctly from one form to another.

Expert Systems

WSD also plays a role in design of Expert Systems and their applications. An Expert system is "A computer program that contains a knowledge base and a set of algorithms or rules that infer new facts from knowledge and from incoming data" (www.dictionary.com). "An expert system is an artificial intelligence application that uses a knowledge base of human expertise to aid in solving problems. The degree of problem solving is based on the quality of the data and rules obtained from the human expert. Expert systems are designed to perform at a human expert level. In practice, they will perform both well below and well above that of an individual expert." (www.dictionary.com)

Expert Systems are created to simulate intelligent behavior to the user and many of them are tested with the 'Turing' Test. The man behind the Turing Test, Alan

M. Turing (1912-1954) named this as "the imitation game" in his 1950 article Computing Machinery and Intelligence which he so boldly began by the following sentence: "*I propose to consider the question "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think."* He proposed a unique approach to testing the validity of expert systems. His work had three attributes to the test namely an interrogator, a human and the expert system to be tested (all in three different rooms). The interrogator queries both the human and the system using a terminal. His/her task is to identify which one is human and which one isn't. If the machine is able to fool the interrogator, then it passed the test. Though this test has been subject to many criticisms, this is one of the most commonly used testing tools of today. Testing algorithms and tools depend largely on external factors like an error free communication, an efficient communication protocol, absence of ambiguity in communication and highly organized flow of the channel. So WSD comes into the fore. WSD makes an expert system perform better. Expert systems are also supposed to learn by themselves which makes it mandatory that at

the time when the data is entering the knowledge base with all tagging and validations, the information should be unambiguous and should let the expert system learn more from future encounters of the context.

Relevance Ranking and Content analysis

Content analysis, per se, is the procedure to analyze the contents of text material to arrive at conclusions like the number of instances of a particular word or a group of words, statistical data inference, sense manipulation, the presence and meanings of different words in the text as related to the author/writer's way of writing and lots of other information used for analyzing and in some cases even evaluating text. For instance a school conducting online courses and examinations might have a tool that analyzes the text and looks for correct answers, even if the answers are essay/paragraph type and not just multiple choice ones. This data may be used by systems like the WSD systems which, in turn, use this data to arrive at conclusions about how relevant is the answer to the sense of the question and evaluating the answers. So WSD forms a major part of content analysis.

Basically content analysis algorithms come with a WSD system concealed within them.

Retrieval of information

Information retrieval(IR) is one of the major applications of WSD systems. Information retrieval (IR) refers to retrieving relevant and related documents from a database or in general a knowledge base. The search engines in the World Wide Web (WWW) are typical examples of such IR applications. WSD systems increase the relevancy of documents retrieved and also ensure the consistency of information. In a situation where the input is unpredictable like in case of a search engine (the query words used are totally dynamic), WSD systems help support in a lot more ways than one.

Statement of Research

This study focused on formulating a unique similarity based concordance approach to word sense disambiguation by enhancing the existing statistical methods using a concordance technique. This research also analyzed and tested the algorithm for increase in the efficiency of disambiguation performance.

Hypothesis

The presence of concordance techniques in probabilistic and statistical algorithms for computation of WSD, increase the accuracy of the disambiguation performance.

Definition of terms

Antonymy: Words with opposite meanings are antonyms, for example, 'rich' and 'poor'. However, it is important to note that [NOT 'rich'] is not the same as ['poor'].

Cognition: The mental process of knowing, including aspects such as awareness, perception, reasoning, and judgment.

Concordance: Agreement and also an alphabetical index of all the words in a text or corpus of texts, showing every contextual occurrence of a word.

Corpus: A large collection of writings or recorded marks of a specific kind or on a specific subject used for linguistic analysis.

Disambiguation: To establish a single grammatical or semantic interpretation for a specific word.

DV: Defining Vocabulary

Hyponymy / hypernymy: Hyponymy and hypernymy relations demonstrate hierarchical categories. For example, 'maple' is a hyponym

of 'tree', and 'tree' is a hypernym of 'beech'.

ICECUP: International Corpus of English Corpus Utility Program. The text analysis program ICECUP was developed to analyze texts annotated with tags specific to the International Corpus of English (ICE).

LDOCE: Longman Dictionary of Contemporary English. This dictionary holds a database of over a 155,000 natural examples of grammar, 1 million additional sentences from books and magazines and top 3000 words in spoken and written English.

Meronymy / holonymy : Represent features of a word for example, 'wall' and 'door' are meronyms of 'house', conversely, 'house' is a holonym for 'wall' and 'door'. These relations are also transitive and asymmetric.

Polysemy: The ambiguity of an individual word or phrase that can be used in different contexts.

POS: Part Of Speech - the attribute of a word in a sentence.

Synonymy: Words with very similar meanings display synonymy. Synonyms must be interchangeable, so words in different "syntactic categories" (noun, verb, etc) cannot be synonyms. This does not mean that similar words in the same syntactic category must be synonyms.

Tagging: A sequence of characters in a markup language used to provide information, such as formatting specifications about a document.

WSD: Word Sense Disambiguation - the process of assigning a specific sense to an ambiguous word from among more than one sense listing.

Assumptions

The following assumptions are made regarding this research:

1. The communication process involves a protocol known to both parties (i.e., the computer and the user).
2. The corpus is free of spelling errors and grammatical errors.

3. The dictionaries make clear and concise distinctions between the senses of a word.
4. The sentences in the corpus make sense with respect to their logic and flow.

Limitations

1. The efficiency of the disambiguation depends mostly upon the ambiguity of the corpus and the words contained in the corpus.
2. In the case of corpuses where the sentences are contextually unrelated, this technique may produce undesired results.
3. This concept of concordance reduces the speed of operation of the algorithm considerably.

Delimitations

The following are the delimitations of this research:

1. This study is restricted to the performance of the sample of ten words used for testing.
2. The execution of the algorithm depends totally upon how the words are placed in each sentence.

3. So, for the same words, the algorithm may produce different results in a different corpus.
4. This research does not use a widely known and standard corpus like the WordNet® due to resource availability constraints.

CHAPTER 2

Review of literature

The automatic disambiguation of word senses has been of concern since the 1950s. Sense disambiguation is an "intermediate task" (*Wilks and Stevenson, 1996*). The earliest approach towards disambiguation dates to the late 40s *(Weaver, 1949)*. It is clear that the question of WSD[1] was raised half a century ago. WSD is obviously essential for language understanding applications like message understanding, man-machine communication, etc. The fact that WSD was of much concern since a long time ago is evident from some examples like:- *sense disambiguation is essential for the proper translation of words such as the French grille, which, depending on the context, can be translated as railings, gate, bar, grid, scale, schedule, etc. (see for instance Weaver,1949; Yngve, 1955.)*. The earliest approaches were the dictionary based approaches which looked for sentence and meaning co-occurrences. The most common dictionary tool used as a knowledge base was the LDOCE[2] The dictionaries used,

---

[1] Word Sense Disambiguation
[2] Longman Dictionary of Contemporary English

though not exhaustive, were a good source of knowledge base for the research. But the question is about the granularity of the sense. Though the dictionaries make clear and concise distinctions between various words and also give various senses for a word, the question arises as to whether the sense returned is useful in this particular context of this particular application. WSD systems therefore have to take into account this issue and work accordingly. Text classifiers form a very important resource for WSD researchers. A Text classifier classifies each word in a given untagged corpus into some category according to related questions called "Queries" in large numbers. A query, in this context, is a form of a question about each word, the answer to which, would help the classifier to categorize or classify the word. Each word is actually analyzed independent of other words with respect to its properties, or in this sense, "attributes". Naïve Bayes classifier is one such classifier used to categorize text.

The Naïve Bayesian Classifier

A text-classifier plays a very important role in the disambiguation process. The Naive Bayes classification is one of the most successful known algorithms for learning to classify text documents. A brief outline of the model would help understanding some of the earliest approaches to WSD. The Naïve Bayes states:

"Let X be the data record (case) whose class label is unknown. Let H be some hypothesis, such as "data record X belongs to a specified class C." For classification, we want to determine P (H|X) -- the probability that the hypothesis H holds, given the observed data record X." *(Cohn 2001)*. P(H|X) is the posterior probability of H conditioned on X. In contrast, P(H) is the prior probability, or a priori probability, of H. Similarly, P (X|H) is posterior probability of X conditioned on H. Where S is the set of senses, and V is the context of the ambiguous word.

$$\begin{aligned} \hat{S} &= \underset{s \in S}{argmax}\ P(s|V) \\ &= \underset{s \in S}{argmax}\ \frac{P(V|s)P(s)}{P(V)} \\ &= \underset{s \in S}{argmax}\ P(V|s)P(s) \end{aligned}$$

$$P(V|s) \approx \prod_{j=1}^{n} P(v_i|s)$$

Now

- Estimate for $P(v_i|s)$ (decrease the probability of previously seen events, so that there is a little bit of probability mass left over for previously unseen events). This step is to ensure that the words are categorized on the basis of probability of their appearance in similar contexts before, if any.
- Estimate for priors - P(s)

The following example illustrates the theorem:

Assume that the data under consideration consists of animals, described by their features and attributes. The native Bayesian classifiers see this data set in this way: "Given an animal that has four

legs, an antler, is a mammal and a herbivore, which type of animal is it most likely to be, based on the observed data sample?. The answer is not very difficult to interpret. So to make the job easier in futuristic interpretation again based on observation, classify a four-legged herbivorous mammal with an antler as that type of animal." An obvious difficulty in this case, of course, comes up when you have more than a few variables and classes. This would require an enormous number of observations to estimate the approximate probabilities.

Naive Bayes classification eliminates the problem requirement of lots of observations for each possible combination of the variables. Here, the variables are assumed to be independent of one another and, therefore the probability that an animal that is a mammal, a herbivore, with antlers and four legs, average 4½" tall etc. and is male will be a deer (except Caribou) which can be calculated from the independent probabilities that an animal is a mammal, that it is a herbivore etc. In other words, Naïve Bayes classifiers assume that the effect of a variable value on a given class is

independent of the values of other variables. This assumption is called class conditional independence, which, is made to simplify the computation and in this sense considered to be "Naïve".

However, bias in estimating probabilities would often nullify the estimated results. But in this case, they do not make a difference in practice because of the fact that it is the order of the probabilities which determine the classifications, not their exact values.

Studies comparing classification algorithms have found that the Naïve Bayesian classifier is comparable in performance with classification trees and neural network classifiers. They have also exhibited high accuracy and speed when applied to large databases. This classifier approach resulted in the development of a new classification approach. Under this approach the classifier generated classification data to the disambiguator, which played the part of a user-dialog processor and fed the disambiguation engine with data

for analysis and classification based on occurrence and coexistence.

Word

Since the word "word" will be used many times and in many contexts, it is useful to look at its meaning, and attempt to relate less ambiguous terms to some of its senses. As *Matthews (1974)* specifies in his book "Morphology", in linguistic terms, "word" has three main senses. Any extrinsic meaning is unimportant. The first sense is where it is represented just as a string of symbols written or spoken and is used as a generic identifier. Any meaning associated with it is unrelated; it is used merely as a "label" in Computer-Scientific terms. The second sense is "the fundamental unit of the lexicon of the language", the base concept from which many words can be derived. The third sense is the most common, which can be described as an "instance" of the second sense. These can have grammatical categories attributed to them, such as noun or verb, and have some meaning and reference point within the language. To disambiguate the terms,

Matthews recommended that the first sense be called "word-form", the second "lexeme" and the third "word" and each word is super-subscripted with the sense number associated with it. Assuming sense1 (when superscripted) represents the sense "word-form", sense2 represents the sense "lexeme" and sense3 represents the sense "word", a word can carry its sense alongside in a sentence. For example, the word-form tried^{sense1} is the form of the word^{sense3} which is called the Past Participle (or the Past Tense) of try^{sense2}. It is more important to distinguish between lexemes and the other two senses - "word-forms" and "words" are very similar in some contexts. However, word-forms can be monosyllabic or disyllabic, but not "nouns", "verbs", etc - these categories are used to describe words or lexemes. There are many relations between words, described by Miller. He classified words under four categories called synonymy, antonymy, hyponymy/hypernymy, meronymy/holonymy.

Concreteness and Abstractness of Words

The LDOCE[3] NLP[4] Database contains definitions which are (primarily) made from words taken from a list of approximately 2000 words - the Longman Defining Vocabulary. Each word in this set was labeled as either "concrete" or "abstract". Concrete words are those which refer to objects, actions, or other sources of sensation directly - these sources can be physically pointed out to someone to show them what the word means. Abstract words are those which refer to objects, actions, or other sources of sensation indirectly - the things these words reference can still be experienced by the senses, though less directly, and as such are harder to point out to someone without some accompanying explanation. When attempting to classify the words in the defining vocabulary as either concrete or abstract, a major problem was encountered - no senses for the words in the list are specified, which implies that all of the word's senses are meant. Sometimes a word would have part-of-speech-specifiers after it, e.g. only the adverb and preposition homonyms

[3] Longman Dictionary of Contemporary English
[4] Natural Language Processing

of "above" are in the DV[5], not the adjective. When several homonyms of a word are present in the DV, the concreteness will vary from homonym to homonym. For example, in the case of the word "back", the noun is fairly concrete, as in "The side of a person's or animal's body that is opposite to the chest and goes from the neck to the top of their legs". However, the adverb explanation is less concrete, as in "In or into the place or position where someone or something was before". So, it becomes difficult even to disambiguate the same word with two different semantically related usages. As each part of speech-type of a word generally correlates with the number of homonyms it has (i.e. the adverb "back" and the noun "back" are two separate homonyms), there are clearly more than 2000 words in the DV.

Just like the problem with different homonyms, there is also a problem with the many senses each word generally has. Looking at the word "back" once again, the noun has 19 senses, and several of these have additional sub-senses. The concreteness of the noun

"back" varies wildly depending on the sense. If it is the case that there is no sense information in the DV (it may be the case that the version examined is incomplete), then simply omitting the words in the DV may not be enough to be able to learn the meanings of the other words in the dictionary. Also there is a need to reason about which sense of the word is being referenced in a particular definition. Research on the DV revealed that the sense numbers of words are indeed not specified in the DV, but only the most common and central meaning of a word is "used", i.e. the words in the DV will not refer to an uncommon sense. In addition, the senses in the dictionary are in frequency order. Generally, the Zipfian distribution is enforced in such cases. This means that if a list of words are in frequency order, the frequency of the second most common word will be half that of the most common word, and the frequency of the third most common will be a third of that of the most common, and so on (*Lesk 1986*). Therefore, it seems reasonable to assume that

---

[5] Defining Vocabulary

the first sense of a word in the dictionary is the one that the corresponding entry in the DV is referring to.

Gorman *(Gorman, 1961)* was one of the first to conduct experiments in which words were either labeled concrete or abstract according to a set of rules. To prepare for these experiments, two "judges" were told to classify a list of words (all nouns) as either concrete or abstract according to the following rules:

- Concrete nouns are "those whose reference to objects, materials or sources of sensation is relatively direct"

- Abstract nouns are "those whose reference to objects, materials or sources of sensation is relatively indirect."

- "A word may be 'abstract' *and* either general or specific, or 'concrete' *and* either general or specific."

- "Classify as 'abstract' all nouns usually classified by grammarians and logicians as abstract in the sense opposed to concrete; also

all nouns that are primarily names of measures, of processes, of kinds of persons characterized by reference to an a-sensory trait (e.g. *optimist*), and any others judged analogous to these."

- "Classify as concrete the names of mythical animals like monsters, and all words judged analogous to these. Disregard any meanings that are judged to be 'unfaded' metaphor (e.g., *gadfly* - a person who irritates others). Apply the same principle of reference to sight, hearing, taste, smell, and somesthesis [(senses which are not localised to specific organs like sight, smell, etc are)]."

- "Assign every word to either the 'concrete' category or the 'abstract' category. Add subscripts where necessary: *m* to indicate that while the word belongs predominantly to one category, some of its meanings belong to the other category, or to indicate that assignment to the category chosen is felt to be uncertain."

These findings point out that Gorman has taken into consideration, homonyms and senses of words. However, homonyms are different words with the same symbolic representation, so it is not enough to say that a symbol belongs predominantly in one category when it refers to several different words.

Other researchers who built on Gorman's experiments (Belmore et al (1982), Holmes and Langford (1976), Klee and Eysenck (1973)) used more than two classes, and also looked at sentences instead of just individual words.

Natural Language Processing (NLP)

Natural Language, in this context means, the language which humans use to communicate with each other. NLP can be briefly described as the use of computers to process written and spoken language for some practical, useful purposes like translating languages, getting information from the World Wide Web and even striking a conversation with a machine. The goal of a Natural Language processing system is to

enable an unambiguous communication between the user and the machine in natural language. This makes the job a lot easier in enabling effective communication with the machines. It is easier for humans to communicate and learn language than it is for a computer because what humans call 'learning', is a behavioral aspect which has to be artificially created in a computer. The challenges mankind faces from a NLP application are worth the research. Compare the understanding of the phrase "Man eating hamburger" against a "man eating shark", by a computer. Is there an algorithm to disambiguate this context? The first question is whether this is possible at all letting alone attempting to solve it. The answer is the "Thinking machine". A very noticeable difference between a human and a machine is the ability to think. So only a Learning Machine can accomplish this task which is why Machine Learning is an important aspect of WSD systems.

Machine Learning

Any natural language processing system involves an effective participation of machine learning systems,

which, in turn, work on statistical data. NLP systems are genetically different from all other algorithmic systems in the sense that they cannot be specified algorithmically. For example, *"how many kinds of living things have three or more legs?"* is a common question, and one can actually sit and count the number whereas there is no algorithm for it. Questions would arise about how the algorithm would look like because of the ambiguity of the real world and continuity of the data set. On the other hand, mere word matching techniques produce highly undesirable results in some cases. In other words, a knowledge base resource may contain the words "living things" and "legs" but still may be unrelated to the question whereas a resource without the words may have related answers. Since Machine learning offers invaluable input to WSD systems, some of the definitions for Machine Learning are discussed in the forthcoming paragraphs.

Learning, like intelligence, covers such a broad range of processes that it is difficult to define precisely. Dictionaries define learning as "to gain knowledge, or understanding of, or skill in, by study, instruction, or experience," and "modification of a

behavioral tendency by experience." Learning is to gain knowledge. Here, the focus is on learning in machines. There are several overlaps and similarities between human and machine learning. It is not inappropriate to say that the concepts and techniques being explored by researchers in machine learning may illuminate some aspects of human learning. When it comes to machines, whenever the structure, or data of a machine changes, it learns in such a way that, the change is used to better the performance of the machine in the future. This machine learning is not identified only in the cases of algorithms where the change in data happens in such a way that it is comfortably placed within the scope of other disciplines and are not necessarily better understood for being called learning. But, for example, when the performance of a WSD System improves after reviewing several samples of text, it feels quite justified in that case to saying that the machine has learnt. A key objective of machine learning is to design and analyze programs that learn from experience.

Sense

Some of the meanings for "sense" include the normal ability to think or reason soundly or a meaning that is conveyed. Sense has also another meaning as in sense of speech. It is seen that when humans communicate with each other, sometimes they mean words in different semantics. The ability to understand the meaning can be explained because they know their context. Consider the case of communicating with a computer which runs a natural language understanding system. Assuming the system uses word matching and frequency analysis techniques to interpret natural language, an input similar to "A stitch in time saves nine" would probably produce an undesirable result set just because of the fact that finding and analyzing commonality among the words is completely different from doing the same with their respective contexts/ senses. The worst case was in early years of NLP when the computer was made to interpret "The spirit is willing but the flesh is weak" and the computer's interpretation was "The vodka is good but the meat is rotten". This shows that many of the existing NLP applications so far have just played with the words in

a sentence and their commonality and not their senses. The machine is programmed to learn the grammar and master the lexicon and then analyze commonality to produce meaningful interpretation. Programs such as ELIZA[6] - the psychologist have seen tremendous successes recently especially with their ability to play with grammar and words.

Word Sense Disambiguation

The most important problem encountered by a Natural Language Processing System is the ambiguity of certain words. Every language has words that are ambiguous in the sense that they might have

a) the same spelling but different pronunciation and/or different meanings.

b) the same spelling and pronunciation but different meanings.

c) more than one usage with different meanings.

For example, consider the sentences

1) The bank was flooded by the water

2) The bank was robbed by a thief last evening.

---

[6] A famous program by Joseph Weizenbaum, which simulated a Rogerian psychoanalyst by rephrasing many of the patient's statements as questions and posing them to the patient

Notice the word bank and its context in either sentence. Solving this problem of semantic ambiguity of words in a sentence is called Word Sense Disambiguation. This ambiguity has been taken care of to some extent by the POS (Part of Speech) taggers which have shown considerable success in recent years.

Knowledge Base

A knowledge base plays a major role in Word Sense Disambiguation. In order for the disambiguation technique to succeed, there has to be a knowledge base that gives vital information such as a data dictionary or a word-sense linker to the disambiguator.

A dictionary definition of knowledge goes:

- The act or state of knowing; clear perception of fact, Truth, or duty; certain apprehension; familiar cognizance;
- That which is or may be known; the object of an act of knowing; a cognition; -- chiefly used in the plural.
- That which is gained and preserved by knowing;

- Instruction; acquaintance; enlightenment; learning; Scholarship; erudition.
- That familiarity which is gained by actual experience; practical skill; as, a knowledge of life; Cognition;"

All these definitions make one thing clear - knowledge is about learning. Experience imparts knowledge. A knowledge base derived from the definition of knowledge can be something which stores knowledge or hosts knowledge. A technical definition says knowledge base consists of: "The objects, concepts and relationships that are assumed to exist in some area of interest". A collection of knowledge, represented using some knowledge representation language is known as a knowledge base and a program for extending and/or querying a knowledge base is a knowledge-based system.

In other words, a knowledge base can be briefly defined as a collection of facts and rules for problem solving. The knowledge base has all the information needed by the application or the user of the application to generate statistics, explain facts and substantiate conclusions. The knowledge base plays a

major part in the system or the application it is associated with because it is the biggest source of information for the respective application. It can also be compared to a storage bin which stores all information that you need in some specified format so that you can access the contents anytime.

Working with a Learning Machine

Machine Learning, is by far, the biggest contributor to disambiguation systems, especially when talking about a statistical approach. Statistical machine learning is a slight variant of machine learning and is a more precise and specific resource to statistical disambiguation systems. Statistical machine learning is different from conventional machine learning systems in the sense that the internal representation is a statistical model, often parameterized by a set of probabilities. For example consider the question of deciding whether the word "watch" is used as a noun or a verb in a given sentence. Anyone who has a mere understanding of the English language would seldom have difficulty in identifying its part-of-speech in a sentence. But how

will a computer do it? One way is to have a collection of sentences some using "watch" as a verb and some as a noun with a label attached to each usage as to specify if it is used as a verb or a noun. The next step would be to invoke a number of machine learning algorithms to bring to life, a *"syntactic disambiguator"* for the word "watch". A rule inferential technique would construct an internal representation consisting of a list of lemmae, perhaps comprising a decision tree *(Berger, 2001)*. For instance, the tree might contain a rule similar to this - *"If the word preceding **watch** is **to**, then **watch** is a **verb**"*. A simple statistical machine learning technique will contain the same rule as well but now equipped with a probability and looks similar to this - *If the word preceding **watch** is **to**, then the probability of **watch** being a **verb** is **p**.* This value p will be arrived at depending upon past documents returned in the same context and the set of sentences in the knowledge base with similar usage.

The task of identifying whether a word in a sentence falls under the category of a verb or a noun or an adjective or any other part of speech is the main question in the approach discussed above. This task is

commonly referred to as the *"part-of-speech* **(POS)** *labeling problem"* which is described as to discover an appropriate set of labels s, one for each of the n words in a sentence. The following is a typical NLP example used in most literature for projecting the part-of-speech labeling problem.

ence	The	quick	Brown	fox	jumped	Over	the	lazy	dog	.
el	DET	ADJ	ADJ	N-S	V-P	PREP	DET	ADJ	N-S	PUNC

Figure 1: POS tagging

Legend:

DET *Determiner*	***PUNC*** *Punctuation*
ADJ *Adjective*	***V-P*** *Verb - Past*
N-S *Noun - Singular*	***PREP*** *Preposition*

In most cases, the word "the" would be a determiner. So life becomes easier when going from the obvious to the ambiguous. But the truth is that such obvious parts of speech can be easily identified and the difficulty lies only in the process thereafter. Because of this difficulty, the earliest automatic tagging systems, based on expert-systems architecture, achieved a pre-word accuracy of only around 77% on the Brown corpus of written English *(Greene and Rubin., 1971)*.The Brown Corpus is a 1,014,312-word corpus of

running English text excerpted from publications in the United States in the early 1960's. The reported number, 77%, refers to the accuracy of the system on the evaluation part of a data set, not used during the construction of the tagger.

It is now definite that the knowledge of any language syntax is not the only aspect which is helpful in creating an accurate tagging system. Beginning with the collection of text (properly annotated with its parts-of-speech), statistical machine learning techniques can be applied to construct an accurate tagger. The Hidden Markov Models (HMMs) are implemented at this point. A HMM is a statistical tool designed and developed to use in robust digital transmission and subsequently applied to a wide range of problems involving pattern recognition. A discrete HMM is an automation which moves between a set of states and produces, at each state, an output symbol from a finite vocabulary. So both the movement between states and the generalized symbols are probabilistic, governed by the values in a stochastic matrix.

A Markov model is a probabilistic process over a finite set, $\{S_1, \ldots\ldots, S_k\}$, usually called its *states*.

The focus is on matters such as the probability of a given state coming up next. A Hidden Markov Model (HMM) is simply a Markov Model in which the states are hidden.

When implementing the HMM in the tagger, the states are the different parts of speech and the output symbols are the words. In producing a sequence of words, the machine passes through a sequence of states corresponding to the parts of speech of the words and at each transition, outputs the next word in the sequence.

Earlier attempts and contributors

Large scale WSD is a complex problem. There were early approaches to WSD like the inference based methods, the specially crafted lexical entries created on a small scale that were developed between the techniques of preference semantics (Wilks, 1978). Most of these including the connectionist approach were quantitative methods and so were limited in terms of implementation as well as conceptualization. "The WSD problem is always denoted as an AI-Complete problem,

that is, the problem of WSD can be solved only after solving all difficult problems of AI like representation of common sense and encyclopedic knowledge" (Nancy Ide and Veronis 1998).

One of the greatest and earliest contributors to WSD was Stevenson. Another major contributor, Yarowsky, worked with small samples, nearly half a dozen words each time because the problem set is huge and the fact that mapping lexical relations of words can be exhaustive, whereas Stevenson solved this problem by linking a large text marked up for WordNet, to a WordNet - LDOCE mapping. Stevenson also contributed the Multi-Engine WSD - a program that learns to combine inputs from a number of sources of lexical information such as preferences (verbs and adjectives), thesaurus (for meanings), topic classes (for subject descriptions) and dictionary definitions. The program also decides which type of lexical information it needs for the specific word. Another important concern would be that any sort of such a disambiguation work involves matching instances of the word with their respective senses in an external knowledge base or with previously disambiguated senses of the word.

One of the earliest approaches to WSD can be traced back to Weaver who referred to automated WSD in the context of Machine Translation. An excerpt from his publication says *"If one examines the words in a book, one at a time […], then it is obviously impossible to determine, one at a time, the meaning of the words. […]. But if one […] can see not only the central word in question but also, say, N words on either side, then if N is large enough, one can unambiguously decide the meaning of the central word. […]. The practical question is : "What minimum value of N will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?" (Weaver 1949).* *Kaplan (1950)* conducted some experiments attempting to answer Weaver's question by proving the hypothesis that sense resolution given two words on either side of the word was not significantly better or worse than when given the entire sentence. Later contributions came from Wilks' performance semantics system *(Wilks 1972).* His system works on semantic interlinks between words and their meanings. His system worked around the assumption that newer usages of words were updated in the lexicon as and when they occurred and claimed that

word-sense and the word have to be identified dynamically. The most obvious limitation of this approach is the fact that the lexicon, however updated, is going to be a limited resource at any moment of time. Some well known approaches to WSD were the Small's word expert approach *(Small, 1980)*, Pustejovsky's Generative Lexicon *(Pustejovsky, 1995)*, Hirst's five part approach to WSD (Hirst, 1987). These were some early findings and early attempts to disambiguate not just words but sentences. In the past ten years, attempts to automatically disambiguate word senses have multiplied to the availability of large amounts of machine readable text and the corresponding development of statistical methods to identify and apply information about regularities in this data. Machine Readable Dictionaries (MRDs), list various words with their synonyms and antonyms along with useful links to each word indicating their relationships with other words. *MRDs contain a rich sense of relationships between their senses and indicate them in a variety of ways (Krovetz et. al 1996)*. The performance of such applications is measured by querying the knowledge base by taking into account \

the set of documents, the set of queries and relevance judgments. Now, the problem of word sense disambiguation has taken center stage, and it is frequently cited as one of the most important problems in natural language processing research today. In contrast, the algorithms to solve the problem have not yet been implemented in real time. The following describes some of the earlier approaches and their advantages and disadvantages.

The Dictionary Approach

Several years after the WSD research began, *Madhu and Lytle(1965)* worked under the observed and proven fact that the domain of the problem largely contributes towards its sense. They calculated sense frequency for different domains and applied a Bayesian formula to determine the probability of each sense in a given context. This method achieved high accuracy and is still a basis for most NLP research. In the mid 80s the concept of MRDs (Machine Readable Dictionaries) came into existence and WSD researchers started using them for their knowledge base sources. Though creating a large lexicon has been a Herculean task, this gave

researchers a confidence in their hypothesis that MRDs form the major source of knowledge base for their research. MRDs certainly have a much larger set of senses for each word than any of the existing knowledge sources, but they aren't exhaustive either. This approach was much criticized, of course, with a major disadvantage with respect to polysemous words (words with multiple meanings). The process of WSD deployed in this approach became biased to disambiguation with respect to senses just described in the MRDs. In other words, the machine doesn't have an ability to learn new senses on its own. Any disambiguation method used onky the words from the dictionary. If the dictionary is not updated with new words and new senses of existing words, disambiguation performance drastically decreases. Lesk(1986) introduced a unique 'signature' concept with his invention of a new knowledge base. The 'signature' has the list of words appearing in the definition of that sense. Disambiguation techniques using his dictionary involved first selecting the sense of the target word. This word's signature contained the highest number of concordances with the signatures of

the neighboring words in the same context. This method achieved 50-70% correctness (*Ide & Veronis 1998)*. Lesk's method contributed a great deal to the MRD based WSD researchers at that time. One other useful algorithm was identified by Bruce and Wiebe in 1994. Here, the informative contextual features were first identified and then out of all possible decomposable models, those that are found to produce good approximation of the data are identified and one of them finally is used to disambiguate.

The Statistical Approach

The problem of WSD was also seen from the statistical point of view. If the system should work upon common sense or even machine learning algorithms, it would become much more complex to implement. One of the other ways to attend to this problem is the statistical aspect. Surrounding words, in a majority of cases, help in the understanding of any text. When one reads a paragraph from some ancient literature writing, which is not understandable after the first reading, human nature is to repeat reading the sentences to determine if the surrounding words or sentences would

be of any help in understanding the context. So, this creates a possible way to come to a solution using statistical techniques. This might not be optimal but is definitely feasible.

The approach of Nancy and Veronis (Veronis, Nancy 1990) suggested the following as some kind of pre-defined senses:

1) A list of senses such as those found in everyday dictionaries.
2) A group of features, categories, or associated words.

These data were used to determine all different senses for every word relevant to the text or the discourse under consideration. But, the problem of assigning appropriate senses to the words still remained a question. Statistical research solved this problem by assigning senses to each word using a first come first served basis and then following induction to propose the actual sense of the word. If one can identify an identical sense of the word in an unambiguous situation of the same corpus, the current situation could be compared to the one which has the

same ambiguous word and had been already identified for its sense.

Table 1: Various senses of the word 'bank'

Sense #	*Training corpus*
SENSE A	The water started flooding the bank.
SENSE A	The bank of the Ohio River is beautiful.
SENSE B	The bank was looted up to $5000.
SENSE B	He is a bank manager.

The algorithm could identify word senses in unambiguous environments where the sense is close to obvious and compare those contexts with the current discourse under consideration.

The Concordance Approach

This approach uses a concordance algorithm (it comes under knowledge based approaches since this thesis involves knowledge based resources such as the LDOCE as a major resource) to come up with a context sensitive word sense disambiguation construct which works on 2 parts - "Agreement of word senses" and "probabilistic"

perspectives. Communication involves collection of information from the preceding words and sentences to determine the current context. This approach works through a series of words and senses from the lexicon and obtains dynamic information to get to the contextual interpretation of the various sentences. In general, the application would consider the following linguistic features:

1) The user's need
2) The context (subject of discussion if there is one)
3) Concepts (words and their properties) of the sentence.
4) Noun-phrases in the sentences
5) Synonyms of various ambiguous words
6) Abbreviation and expansion
7) Misspelled or misspoken words

In general, approaches to WSD have been classified into three types - Knowledge based, Corpus based and bootstrapping *(Mark Stevenson, 2002)*. Most of the existing systems use the knowledge based approach or the corpus based approaches depending upon the specifics of the problem set. They also work in

conjunction with the statistical and probabilistic practices to attempt a solution.

Chapter 3

RESEARCH METHODS

The Disambiguation Procedure

The algorithm used for this research is an extension of Yael Karov and Shimon Edelman's iterative algorithm (Karov, Edelman 1996) to assign number senses to a word in a sentence. The probabilistic approach proposed by them has been widely used in recent years and at the starting point of this new approach. The algorithm is blended along with a concordance approach to increase the success rate and enable better disambiguation with the concordance properties used as a booster. The entire operation was manually traced due to some constraints on implementation time and resources.

This approach employs the word similarity disambiguation (Karov and Edelman 1996) at the first step of execution. Research showed significantly consistent results in the process of execution and it has been an important source of statistical WSD tasks. However, the algorithm does not employ WordNet anywhere in its execution. The initialization of the word similarity matrix using WordNet (Miller et al., 1993) may seem to be advantageous over simply setting it to

the identity matrix, as Karov and Edelman hypothesized. The proposed approach in this thesis focuses on enhancing the performance of the algorithm by using a concordance or word matching technique within this system. After every iteration in the disambiguation process, the score obtained is compared with the contexts of the previous two sentences. This makes sure that the sentences currently in the process are disambiguated again to be tested with the current word and context under consideration. So, before the final disambiguated sense is being returned, the algorithm also makes sure that the concordance properties are checked. This increases the time of execution considerably since there is a recursive process involved but nothing could be predicted about the actual effects in real time with high capacity processors.

The aim of this approach is to make sure it disambiguates the appearances of a polysemous word W with senses $S_1 \ldots S_k$, using the appearances of W in an untagged corpus as examples of previous occurrences in the same context. Due to the resource availability and implementation constraints, this research was traced

for its performance manually. So the training examples were tagged manually, which, by far, has been the most difficult aspect of this research. After each iteration, what is added on to the training set, is a set of additional sense-related examples, which is called an 'update' *set*. This update set for sense s_i of word W is the union of all contexts that contain some noun found in the entry of $s_i(W)$ in a MRD[7]. The feedback sets can be intensified, in turn, by original training-set sentences that are closely related to one of the feedback set sentences; these additional examples can then attract other original examples.

The feedback sets constitute a rich source of data that are known to be sorted by sense. Specifically, the feedback set of s_i is known to be more closely related to s_i than to the other senses of the same word. Dependency is upon this observation to automatically tag the examples of W, as follows. Assign each original sentence containing W to the sense of its most similar sentence in the feedback sets. Two sentences are considered to be similar insofar as they contain similar words (they do not have to share any word);

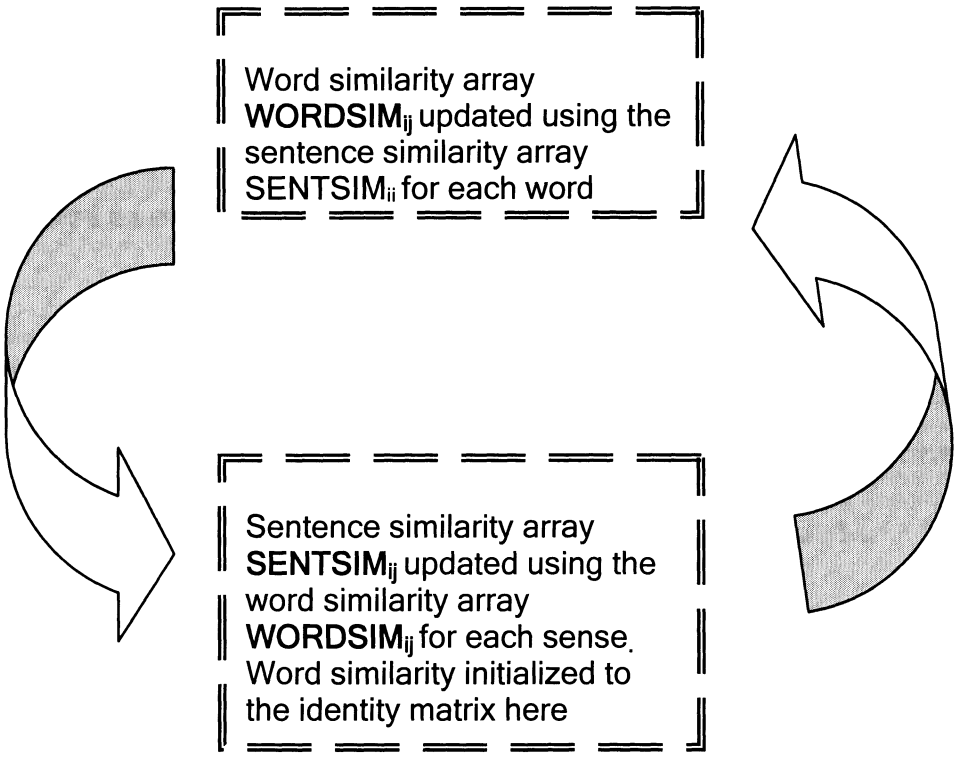


Figure 2 Recursive array updation using the concordance method.

Now we use what is called the "affinity" of the words (*Karov, Edelman 1996*). Updating the similarity matrices involves a procedure where auxiliary relation between words and sentences (fig 2), which is called *affinity*, is introduced to simplify the symmetric iterative treatment of similarity between words and sentences. A word W is assumed to have a certain simpatico to every sentence. In other words, affinity generally reflects the contextual relationships between W and the words of the sentence. This makes it clear

[7] Machine Readable Dictionary or a Thesaurus or any combination of such knowledge

that affinity is a number which represents a real number between 0 and 1 (since relationships between two quantities cannot be less than 0 and greater than 1). If W totally belongs to a sentence S, its affinity to S is 1; if W is totally unrelated to S, the affinity is close to 0; if W is contextually similar to the words of S, its affinity to S is between 0 and 1. Symmetrically speaking, a sentence S has some affinity to every word, reflecting the similarity of S to sentences involving that word. One could use the notation 'a word *belongs* to a sentence', denoted as W ∈ S, if it is textually contained there. In this case, sentence S is said to *include* the word W: S ∋ W. Affinity can be mathematically defined as follows:

$$\mathrm{aff}_n(W, \mathcal{S}) = \max_{W_i \in \mathcal{S}} \mathrm{sim}_n(W, W_i) \qquad (1)$$

$$\mathrm{aff}_n(\mathcal{S}, W) = \max_{\mathcal{S}_j \ni W} \mathrm{sim}_n(\mathcal{S}, \mathcal{S}_j) \qquad (2)$$

where n denotes the iteration number. Now, the sentence, instead of being represented as a mere collection of words, is being represented as a similarity group. Every word has some affinity to the sentence, and the sentence can be represented by a

vector indicating the affinity of each word to it. Similarly, every word can be represented by the affinity of every sentence to it.

Moreover, $\text{aff}(\mathcal{S}, \mathcal{W}) \neq \text{aff}(\mathcal{W}, \mathcal{S})$, because W may be similar to one of the words in S, which, however, is not one of the topic words of S i.e., it is not an important word in S. In this case, $\text{aff}(\mathcal{W}, \mathcal{S})$ is high, because W is similar to a word in S, but $\text{aff}(\mathcal{S}, \mathcal{W})$ is low, because S is not a representative example of the usage of the word W *(show reference)*.

The similarity of word W1 to word W2 is specified to be the average affinity of sentences that include W1 to those that include W2. The similarity of a sentence S1 to another sentence S2 is a weighted average of the affinity of the words in S1 to those in S2. This relationship is represented as follows:

$$\text{sim}_{n+1}(\mathcal{S}_1, \mathcal{S}_2) = \sum_{\mathcal{W} \in \mathcal{S}_1} \text{weight}(\mathcal{W}, \mathcal{S}_1) \cdot \text{aff}_n(\mathcal{W}, \mathcal{S}_2)$$

$$\text{sim}_{n+1}(\mathcal{W}_1, \mathcal{W}_2) = \sum_{\mathcal{S} \ni \mathcal{W}_1} \text{weight}(\mathcal{S}, \mathcal{W}_1) \cdot \text{aff}_n(\mathcal{S}, \mathcal{W}_2)$$

where the total of the weights is 1. It is very important here to note that the weight of a word

estimates its expected contribution to the disambiguation task, and is a product of several factors: the frequency of the word in the corpus, its frequency in the training set relative to that in the entire corpus; the textual distance from the target word, and its part of speech. Initially, all the sentences that include a given word are assigned identical weights.

Initially, only identical words are considered similar, so that $\text{aff}(W, S) = 1$ if $W \in S$; the affinity is zero otherwise. Thus, in the first iteration, the similarity between S1 and S2 depends on the number of words from S1 that appear in S2, divided by the length of S2 (note that each word may carry a different weight). In the subsequent iterations, each word $W \in S_1$ contributes to the similarity of S1 to S2 a value between 0 and 1, indicating its affinity to S2, instead of voting either 0 (if $W \in S_2$) or 1 (if $W \notin S_2$). Word similarity is enhanced significantly by sentence similarity. An example would demonstrate how the similarity based concordance approach discussed above will be effective.

Consider the three fragments

Fragment F1 : drink water

Fragment F2 : pour water

Fragment F3 : drink cola

Here there is no similarity between the words 'pour' and 'cola' when you consider the fragments F2 and F3 under normal similarity based concurrence systems. This is mainly because the context set of these two words is different.

The algorithm used in this study would identify the similarity between fragments F1 and F2 to be 0.5 and the one between F1 and F3 to be 0.5 as well. Here it identifies two relations:

- 'water' is similar to 'cola' because of the usage similarity between 'drink water' and 'drink cola'.
- 'drink' and 'pour' are similar because of the usage similarity between 'drink water' and 'pour water'

Now, 'pour water' and 'drink cola' are similar because in the previous step, there was some similarity between 'water' and 'cola' and some similarity between 'pour' and 'drink'. This relationship is arrived at in

the second step and this relationship holds true if and only if the previous step yields a relationship which can be used to infer such a result after execution. However, there is one big concern: the question whether this relation is asymmetric or symmetric. That complicates the results significantly and the disambiguator may end up in unexpected comparisons and relationships if not properly structured to handle such property of words. The relationship resulting from the execution of the aforesaid algorithm is asymmetric. For example, 'computer' is less likely to be similar to 'monitor' than 'monitor' is to 'computer'. Similarly sentence similarity is also asymmetric i.e., if sentence S_1 is contained in sentence S_2 then $sim(S_1,S_2)=1$ whereas $sim(S_2,S_1)<1$. Each sentence in the training corpus is assigned the sense of its most similar sentence in one of the 'update' sets of sense S_i, using the final sentence similarity matrix. But before this step and after each iteration, the final senses are compared with another similar matrix which hosts the various senses encountered in the previous

two sentences. This comparison is assigned values from 0 to 1 just like any other typical comparison qualifier. Once this stage is carried out, the next step, however is to assign the sense to the word and update the matrix.

The algorithm was tested with sample data and the results were significant. Such a matrix representation for similarity based words and sentences where the update sets are refreshed at runtime, has proved to be very helpful in accomplishing the task of disambiguation. Though employing the update sets' during the execution of the algorithm makes the performance potentially lower in terms if optimality of time complexity, the benefits take over the time trade off.

Chapter 4

Results and Discussion

The algorithm was tested on the ICECUP[8] corpus with over 20000 words considering 10 words which are polysemous. The average success rate of this algorithm was 89%. The original training set (before the addition of the feedback sets) consisted of a few dozen examples, in comparison to thousands of examples needed in other corpus-based methods (Schutze, 1992; Yarowsky,1995). Results obtained from an initial sample set are tabulated in Table 2.

Table 2: List of words used for this study and their corresponding test results

Word	Senses	Sample size	Update set size	% correct (comprehensive)
Bat	Sports equipment	30	67	
	Animal	10	103	92.5
Nail	Part of the body	20	150	
	Piece of metal	5	50	78.2
Ring	Ornament	16	122	
	Phone ring	4	100	89.1
Bank	Financial Institution	18	56	
	River bank	7	78	73.3
Advance	Move forward	10	106	
	Before	38	79	75.9
Change	transform	28	69	
	money	10	102	56.7
Crop	Cultivated plants	156	198	
	To cut	17	97	86.6
Issue	To pass	78	154	
	Dispute	90	109	78.4
Light	EM radiation	69	154	
	To burn or kindle	167	133	
	Not dark	19	80	78.3

[8] International Corpus of English Corpus Utility Program

Charge	Impose monetary penalty	15	145	
	Attack	37	56	59.7

The following are the results of running the experiment on the word "BAT". A graph showing the points in performance against iteration number is shown in figure 3. The success rate of each sense is plotted, and for the weighted average of both senses considered. The marked points in triangles represent the performance points of the sport equipment sense of the word on a weighted average method.

For each example S of the *sports equipment* sense of bat, the value of $sim^n(n,S)$ appears to increase with n. A very important point to note here is that after a minimum of eight iterations the similarity values are closer to 1, and because they are bounded by 1 (similarity can only be between 0 and 1), they cannot change significantly with further iterations. Table 3 shows the performance of the experiment with the word 'bat':

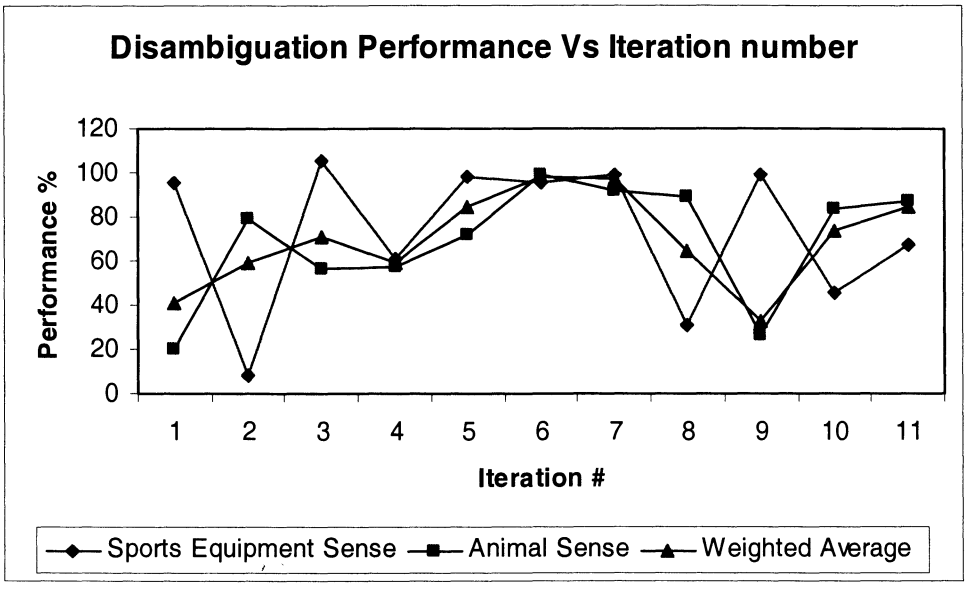


Figure 3 - The 'bat' Experiment

Table 3 shows the most similar words found for the words with the highest weights in the '*bat*' example. One important issue is that the similarity totally depends on the context, and is totally affected by the target word. For example, 'ball' was found to be similar to '*stadium*', because in 'bat' contexts the expressions 'bat and ball' and 'stadium' are highly related. Obviously, 'ball' and 'stadium' need not be similar.

Now, the values were plotted on a graph showing each iteration and the sense of the word that had a higher degree of probability (which is claimed to be the sense of the word in that particular iteration).

The word senses specified in Table 3 show the related words that were identified by the algorithm.

Table 3 : The word 'BAT' and related words in two of its senses

Word : BAT	
Sense	**: Sports Equipment**
Similar words derived by the algorithm	
Ball	: baseball, football, game, play, win, lose
Game	: ball, bat, win, lose, chance,bet,audience
Stadium	: game, ball, bat, flyer, run, catch
Public	: audience, common, people, general
Watch	: game, see, time, when, careful, movie
Pizza	: eat, party, game, football, topping
Baseball	: batter, pitcher, swing, catcher, infield
Bowl	: ball, spin, pace, pitch, Yorker, stump
Sense	**: ANIMAL**
Similar words derived by the algorithm	
Wings	: bird, fly, eat, beak, air
Animal	: legs, hands, eyes, blood, life
Experiment	: animal, species, sample, night
Mammal	: flying, offspring, isolation

This sentence was taken as an example to demonstrate the execution of the algorithm -"Suddenly, the bat flew from the ground and landed on a spectator leaving him in a pool of blood dripping from the head and left him unconscious, after which the match was canceled."

During the first iteration, the word 'flew' (flying) sense didn't do much help to the disambiguator making the chances of the sense being the 'animal' a bit higher than the 'sports equipment'(see Figure 4). The update sets didn't reflect any sense for the word in the context of a game or a match. But during the second iteration, the word 'spectator' gave a little bit of a higher probability to the sense 'sports equipment' to the word and gave it better chance of being the sense for the word we are trying to disambiguate. After the first iteration the similarity of the sense being 'animal' was 0.16 against the probability of the 'sports equipment' being 0.14 (See Figure 4). Though the difference in numbers represented as probabilities is minimal, the execution shows that the sense is identified iteratively over a period of time. After the second iteration, the probability of the 'sports equipment' slightly increased since some

more words were found similar to the sense described in the current context and thus after the third iteration, the 'sports equipment' had a possibility of 0.83 over the 'animal' sense which had 0.79(Figure 4).

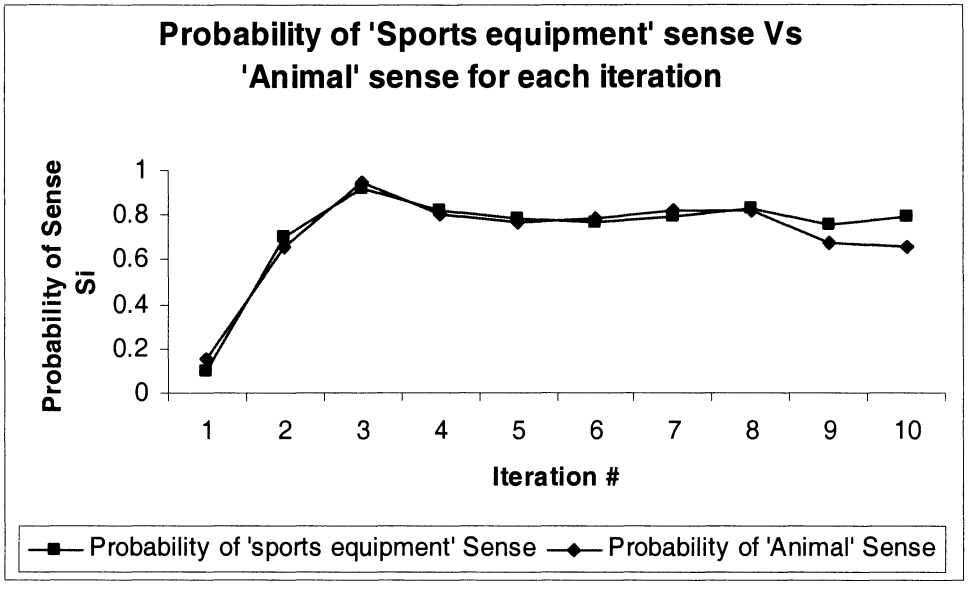


Figure 4 : Results of running the algorithm on the word 'bat'.

The algorithm was tested similarly with a bigger and a better corpus where the words are wide enough with respect to sense and the sentences were close enough in meaning to further complicate the disambiguation process. Though the algorithm initially showed little difference in probability of the two senses, it was observed that significant difference was seen after the 6^{th} iteration. This number 6 represents

the minimum number of iterations this algorithm had to execute recursive calls to identify differences in senses of words. It may vary for other corpuses since the number, placement and frequency of words differ across various corpuses. The algorithm was observed to perform better after the introduction of the statistical update sets and the probabilistic sense matching. The results of executing of the algorithm on each of the other nine words sampled are shown in Appendix A. The words are nail, ring, bank, advance, charge, crop, light, change, issue.

After the algorithm was tested on 10 ambiguous words in a huge corpus, it was seen that the performance increased drastically after the inclusion of the sentence comparison and the backtracking capabilities. Throughout the execution of the algorithm for the various words listed, it was found that the average iteration number at which the algorithm separates the two senses approximately correctly was 8(*as seen from the diagrams in Appendix 1 and Appendix 2*). So, basically, the correctness of the results start appearing only from the 8^{th} iteration and onwards. Some words took longer to be disambiguated than others.

Moreover for some words like 'change', 'crop' etc., where the meanings are close enough, the performance was not satisfactory. But the difference in the probabilities can be seen clearly from the 8th iteration onwards. Almost all other words were disambiguated to their closest senses.

During the execution of the algorithm on the word 'advance' (Appendix A ; Appendix B), the performance was not as expected for more than 4 or 5 iterations and it was seen that though the sentences were consistent to their neighboring sentences, the disambiguation performance was not affected until the words in the neighboring sentences were also encountered in the loop to be disambiguated. One more example was the word 'bank'. Typically, these words were some of the most difficult ones in the corpus to be disambiguated. Since this whole algorithmic process was manually traced, disambiguating words like these was the least optimal in terms of time.

In some cases, like the word 'change', the algorithm performed well and it was showing disambiguation results in the 6th iteration itself but again on the 8th and 9th, the performance went down

again till the 10^{th} iteration where the program would stop execution. This shows that disambiguation is totally dependent on the formation of the sentences and the presence of the specific word (to be disambiguated) in those sentences. An optimal solution also would depend upon such parameters making the process all the more difficult for certain words.

Most of the words used in this research were much ambiguous than others in the category of ambiguous words. The sample was purely a random sample of ambiguous words. The words were picked from the LDOCE at random arranging the ambiguous words in an array and using a random number generator for the array index.

Overall, the algorithm worked effectively for most cases (7 out of 10). Though the sample was relatively small due the unavailability of some resources including time, the algorithm showed considerable improvement over its predecessor. The overall average performance of the algorithm was 70.0%. This number represents the performance of the algorithm after the inclusion of the concordance and statistical techniques. Optimality of time was not used as a measure to calculate performance since the emphasis was

on correct results. Thus, it was observed that, the performance of the disambiguation process is considerably aided by the inclusion of concordance and statistical techniques.

Chapter 5

Summary and Conclusion

In the first stage, tests were conducted on a single word and the concordance techniques proved to enhance the operation. In a later stage, many other words were picked from the LDOCE and were tested with this algorithm on a larger corpus. The outcome depended mostly upon the ambiguity of the corpus, the placement and frequency of the words used. It is also noted that though the algorithm can work around on highly ambiguous words to disambiguate them at some iteration number when there is a significant difference between the two senses, it may not hold true for all cases, needless to say it cannot be a "perfect" algorithm to disambiguate every word it comes across. One major drawback is that this algorithm does not use a widely known corpus like the WordNet® due to resource constraints, which makes its success quite questionable among others in the same category. One other limitation is that this concept of concordance, reduces the speed of operation of the algorithm, obviously due the fact that it has to get into more recursions before going from one word to another and the worst case is when two

sentences with the most number of words, follow each other. Here the algorithm takes a lot more time than even the time taking normal execution. Time efficiency was traded off for better performance. A lot of time is being spent with the drill down process. The results were definitely worth the trade off.

This framework for WSD resolution has some advantages with respect to the fact that though it works around some assumptions, it gives a syntactical and semantic search of senses which will be to the complexity of $O(\log n)$ where n is the number of senses of the word (in case of words which are highly polysemic, the running time of this algorithm varies and usually takes a longer time to disambiguate words). But, it should work for small values of n very well. The probabilistic property of the algorithm makes sure that the algorithm is on the right track with the previously identified senses. Speech recognition, Speech processing, Machine Translation and Natural Language processing, sense retrieval, information retrieval, relevance ranking are some key words in this research as well as some areas where the implementation of this study would be appropriate.

Chapter 6

Future Recommendations

The performance of this algorithm can be tested using a better and widely known corpus such as the WordNet®. Considerable success on such a corpus would pave the way for future research in the area. An implementation of this research would be one of the best future recommendations at this stage since it would make the entire testing process much less difficult. It will also allow testing a larger sample like more than 100 or 150 words which is a considerable number when considering the time taken to run each word.

Another useful research area from this study would be the relevance ranking in web pages on the internet. Relevance ranking is when a search engine on the internet needs to find out how relevant a particular page is, to the queried word or phrase, before it outputs to the searcher. A similar implementation would be the automatic grading of answers in paragraph or short answer form for example. A machine could be programmed to grade the answers using word matching, sense matching, relevancy techniques.

Speech processing and speech recognition are also some research areas where this approach would be very useful. Natural language communication with machines would be one of the most important tasks ahead in the field, which is why WSD finds a place. On the other hand, simulating human language and forms of communication is a developing area of research.

Appendix A

Experiments on each word

Figure 5) Word : Advance

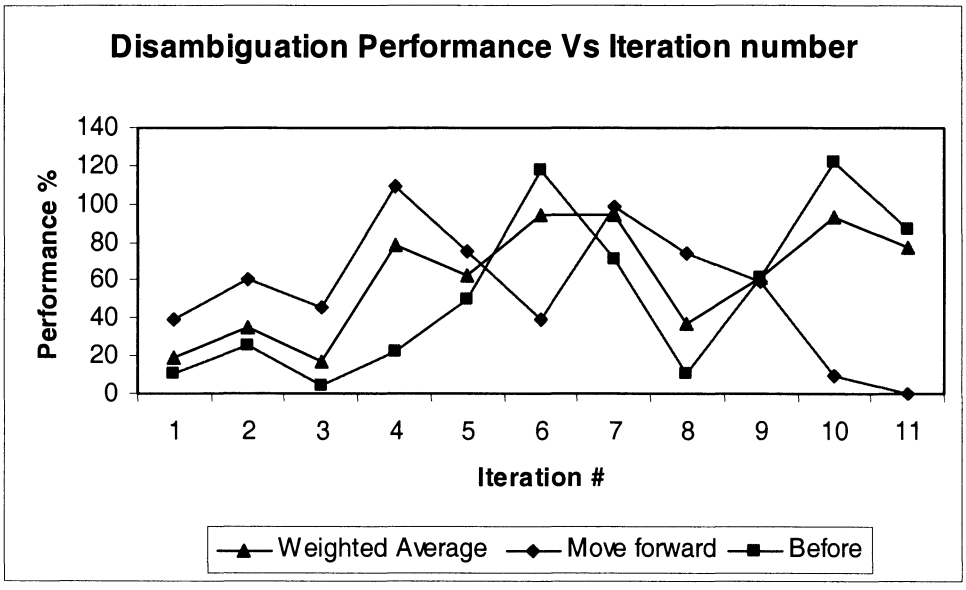


Figure 6) Word : bank

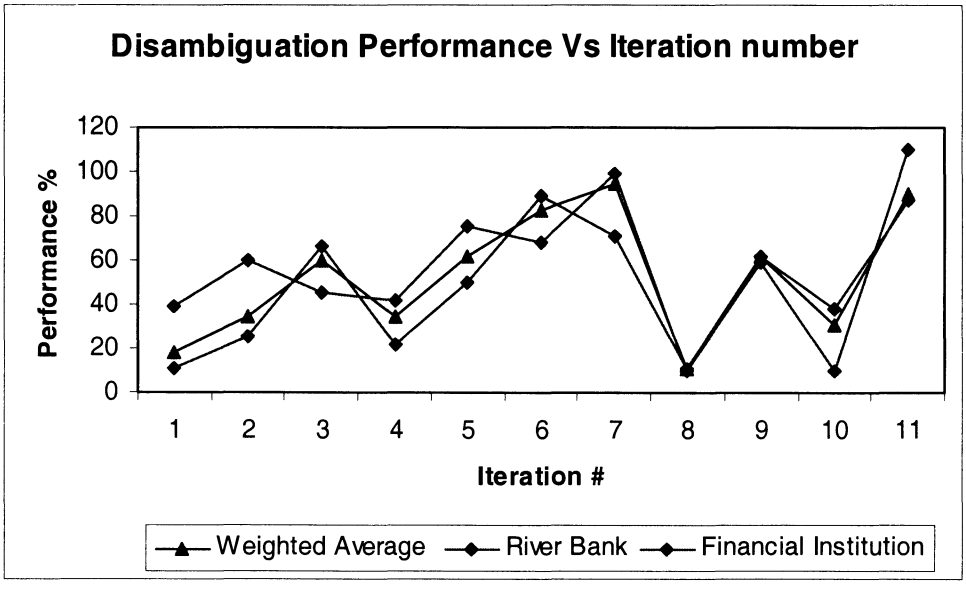


Figure 7) Word : change

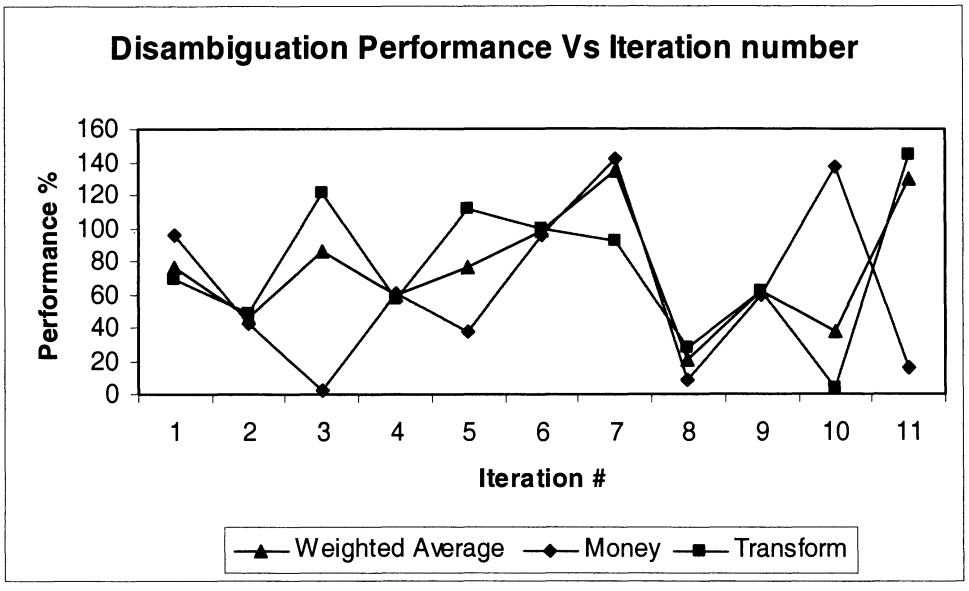


Figure 8) Word : Charge

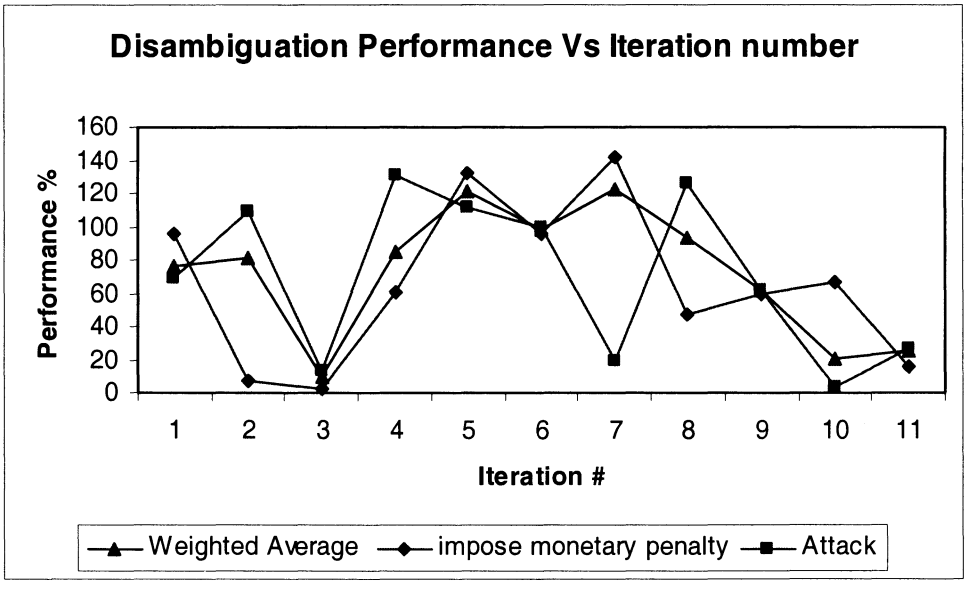


Figure 9) Word : crop

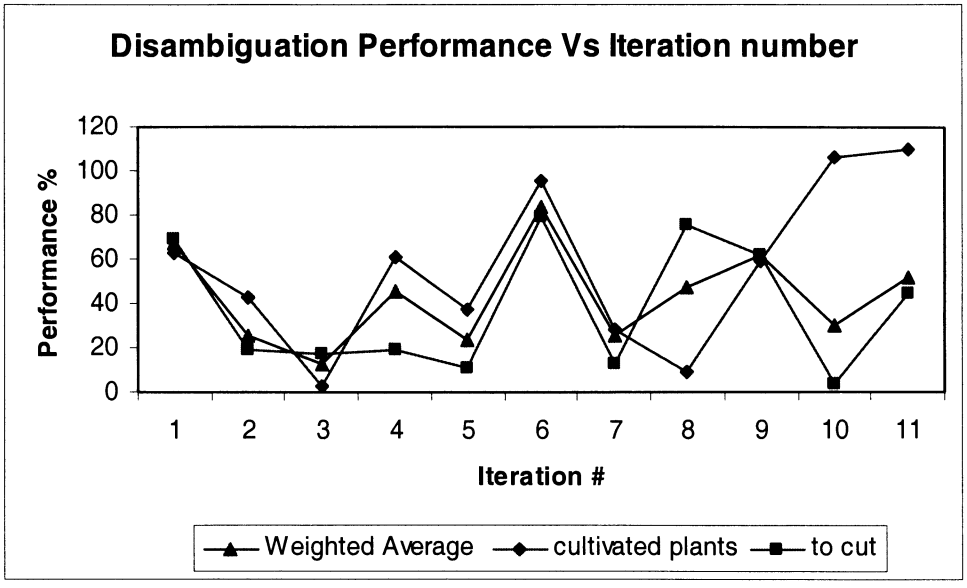


Figure 10) Word : Issue

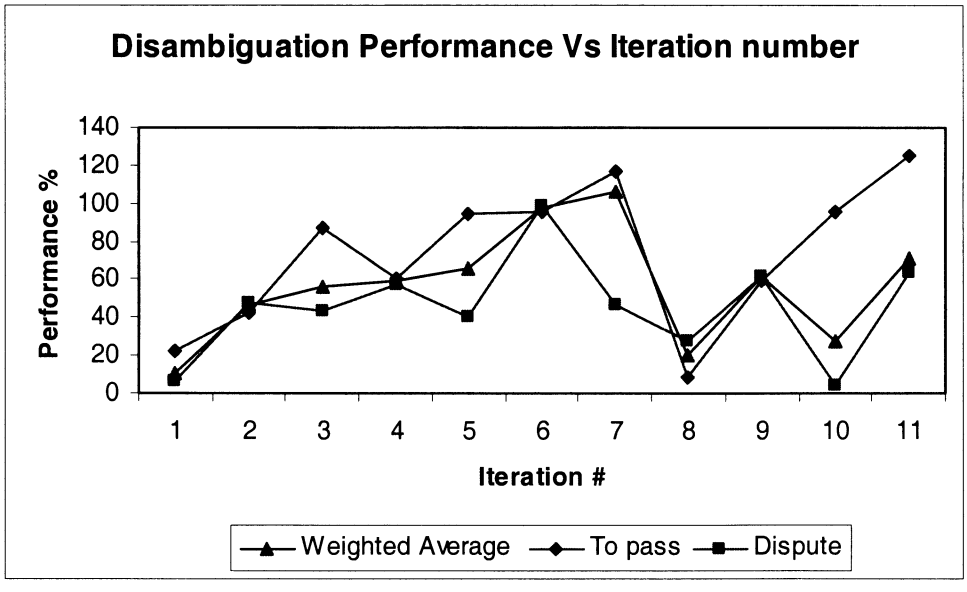


Figure 11) Word : light

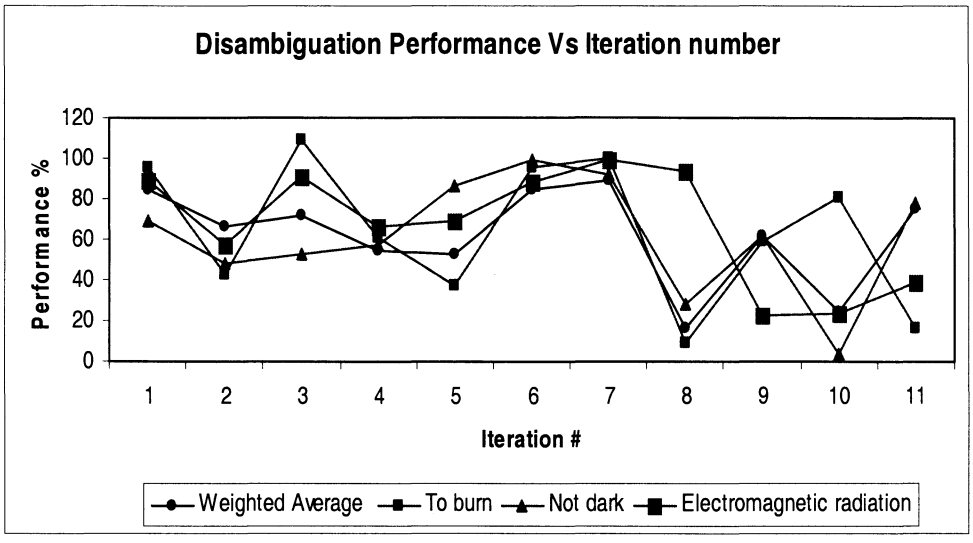


Figure 12) Word : nail

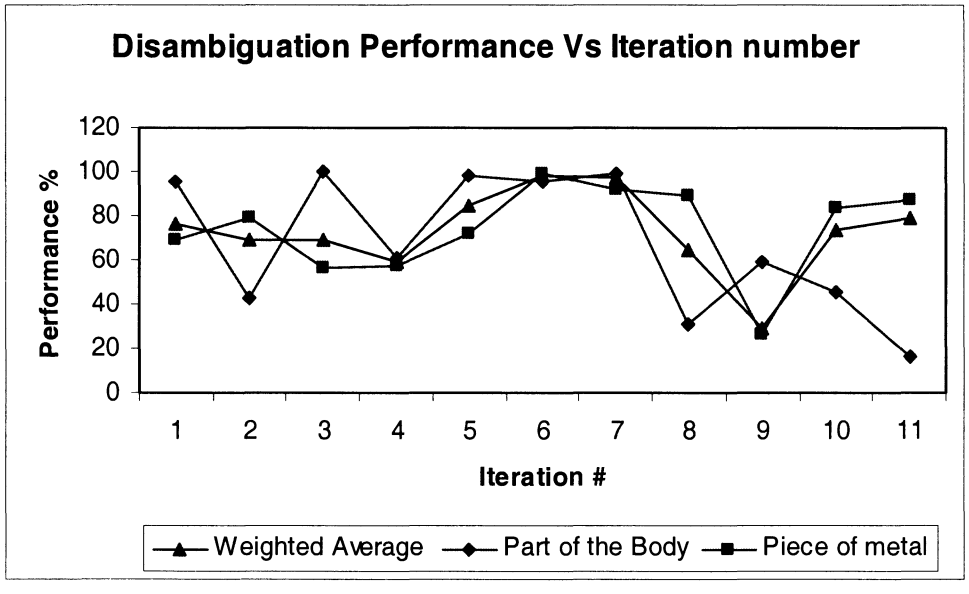
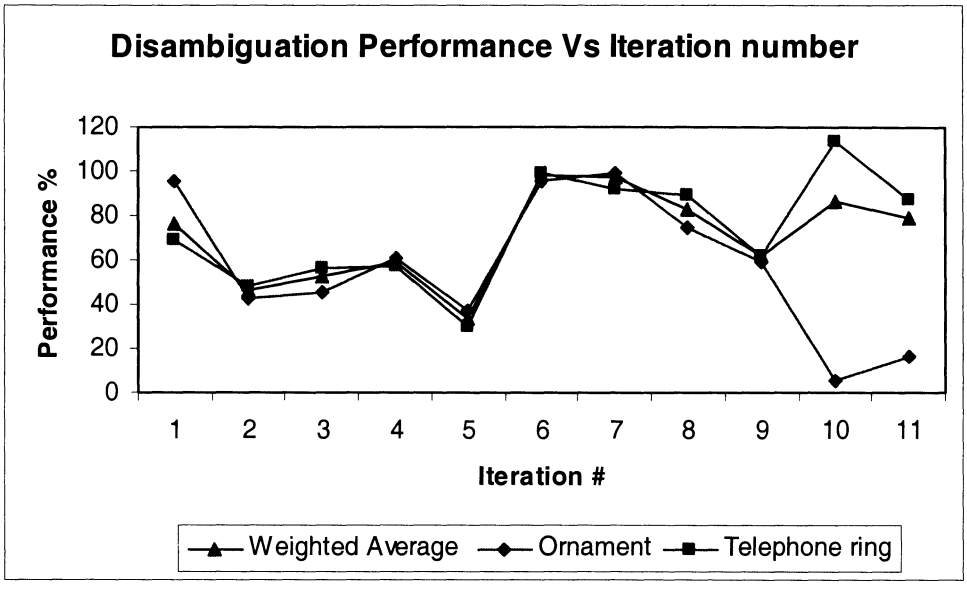


Figure 13) Word : ring



Appendix B

Results of running the algorithm on each word

Figure 14) Word : Advance

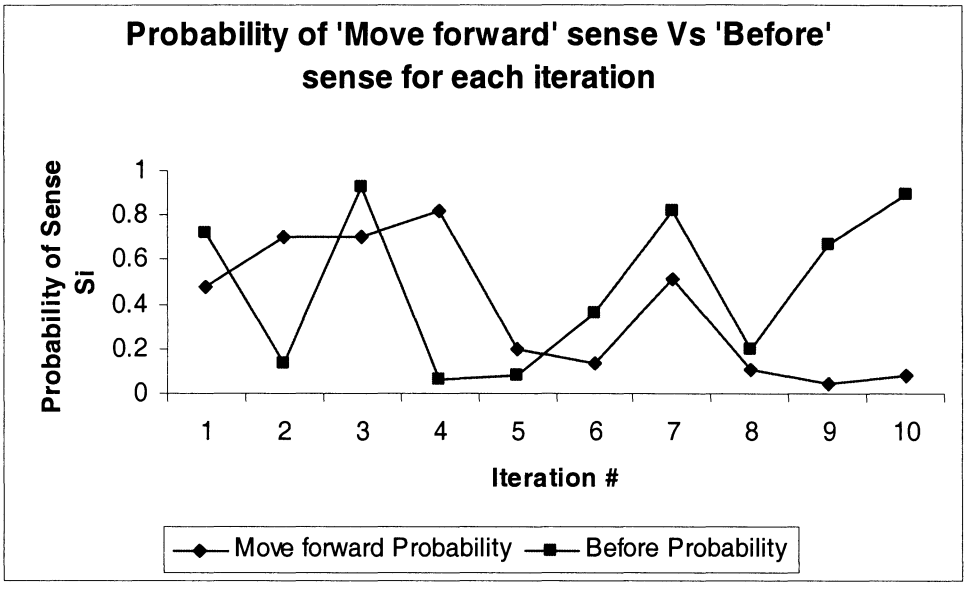


Figure 15) Word : Bank

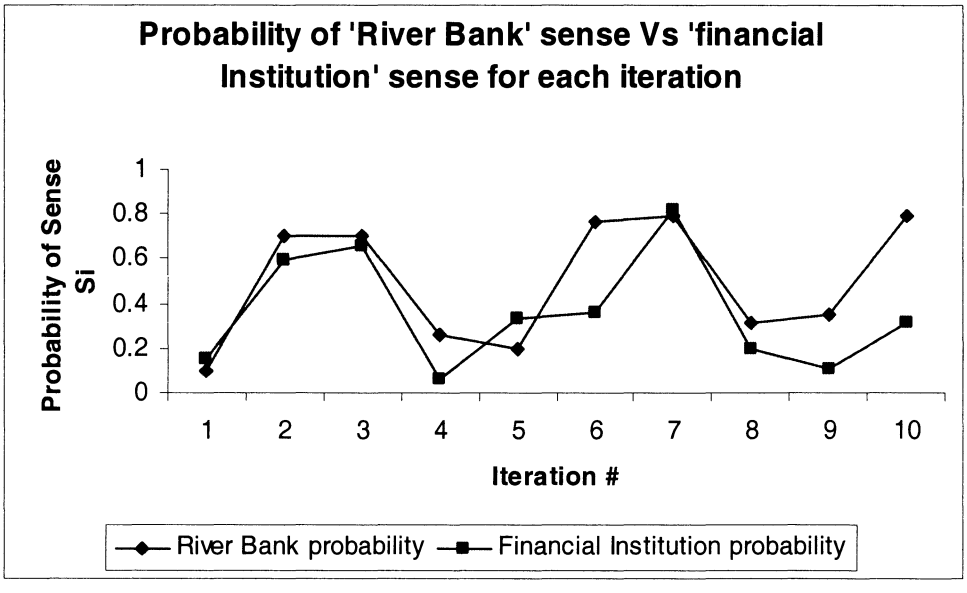


Figure 16) Word : change

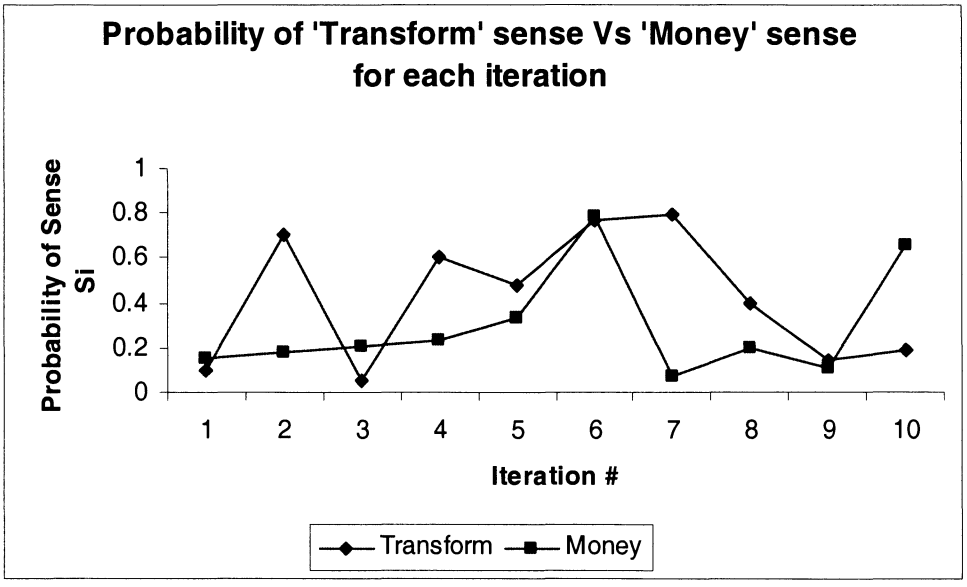


Figure 17) Word : Charge

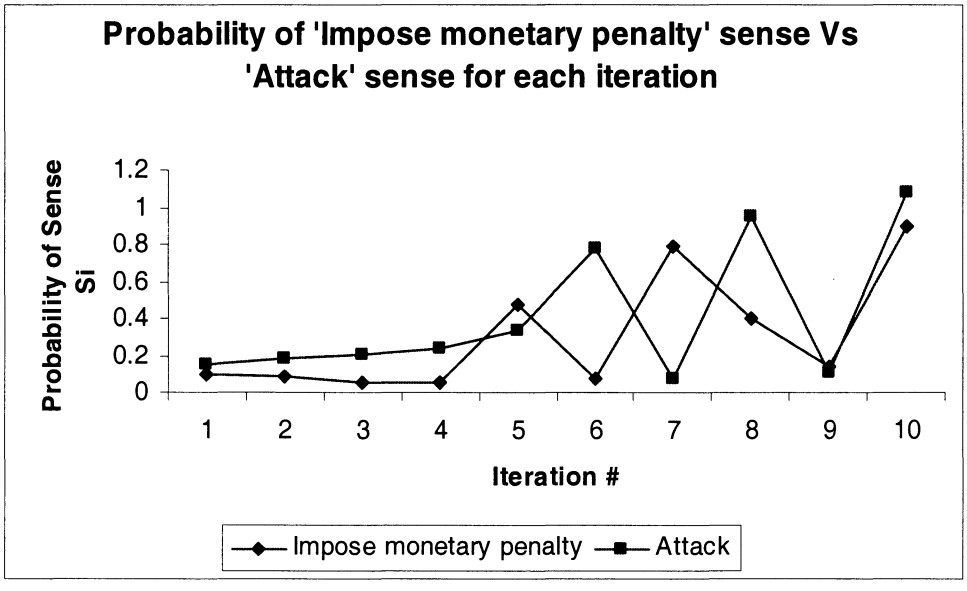


Figure 18) Word : crop

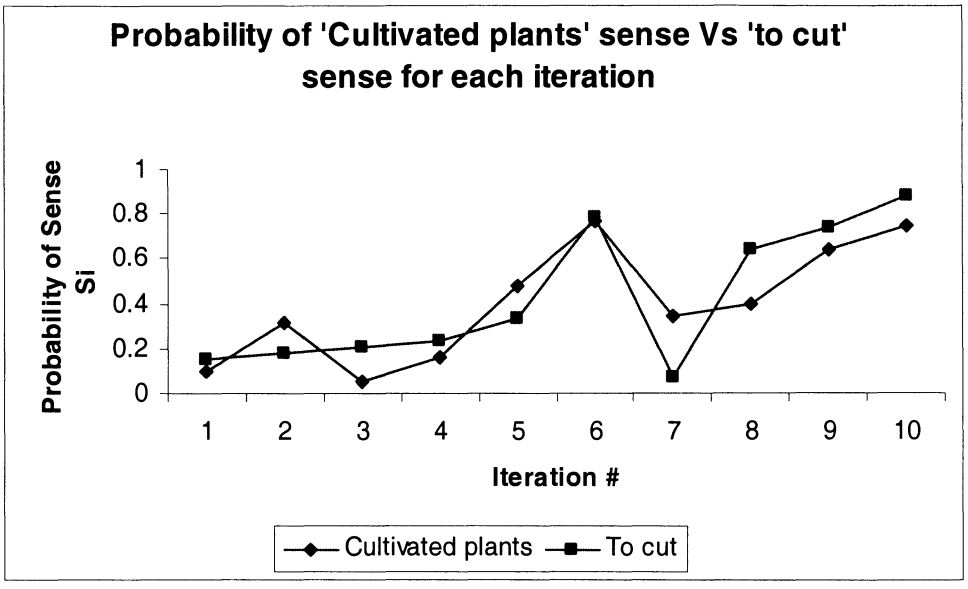


Figure 19) Word : Issue

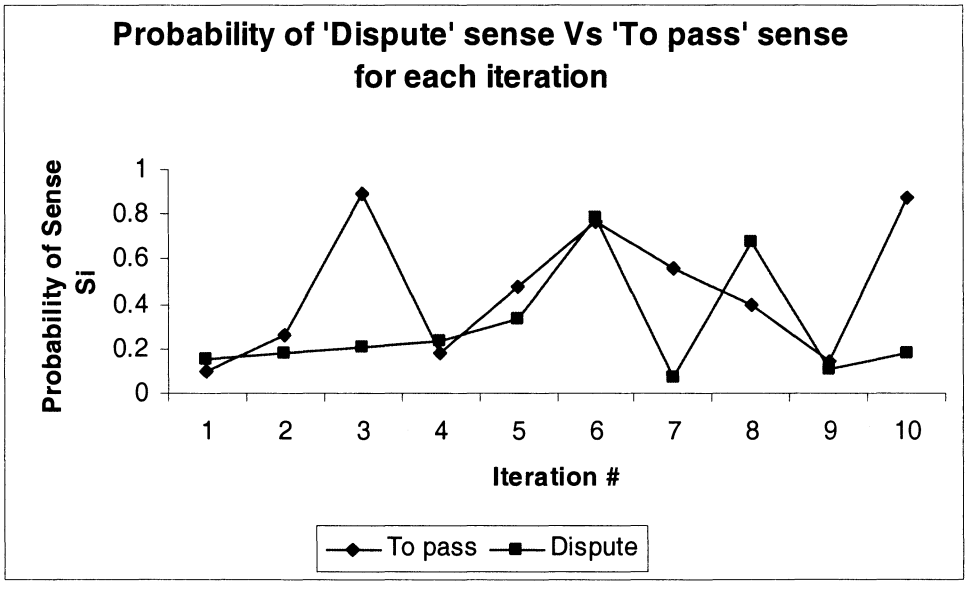


Figure 20) Word : light

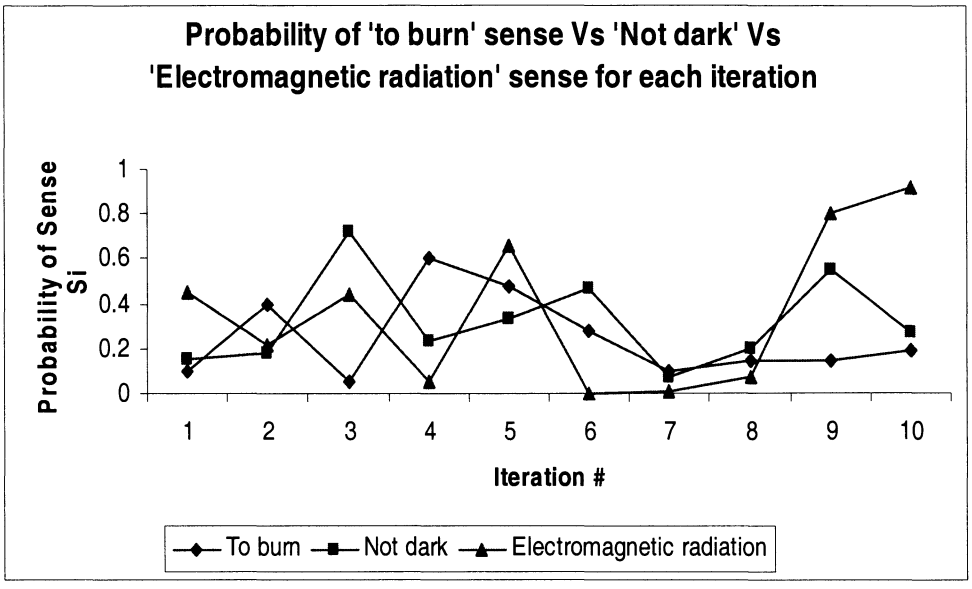


Figure 21) Word : Nail

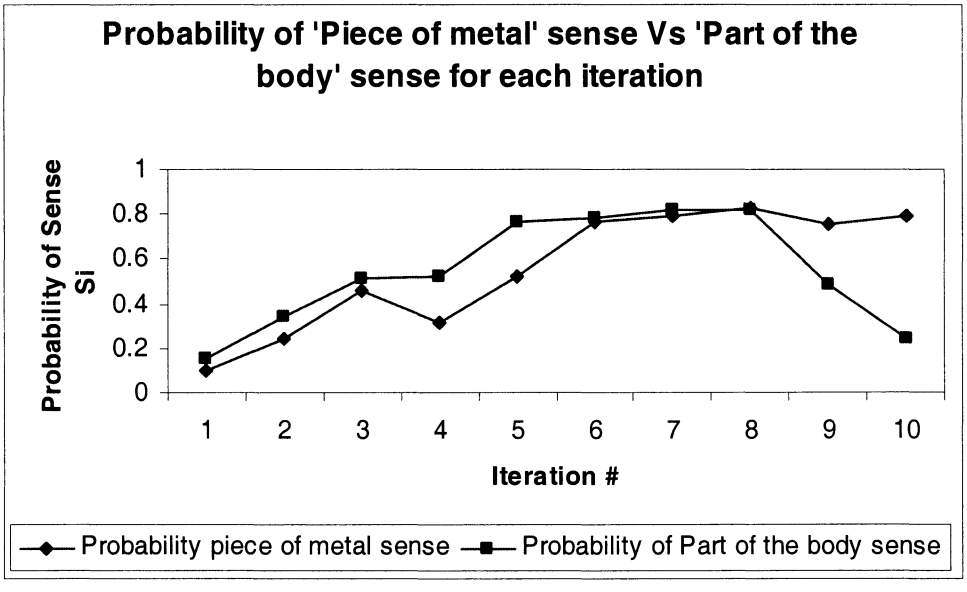
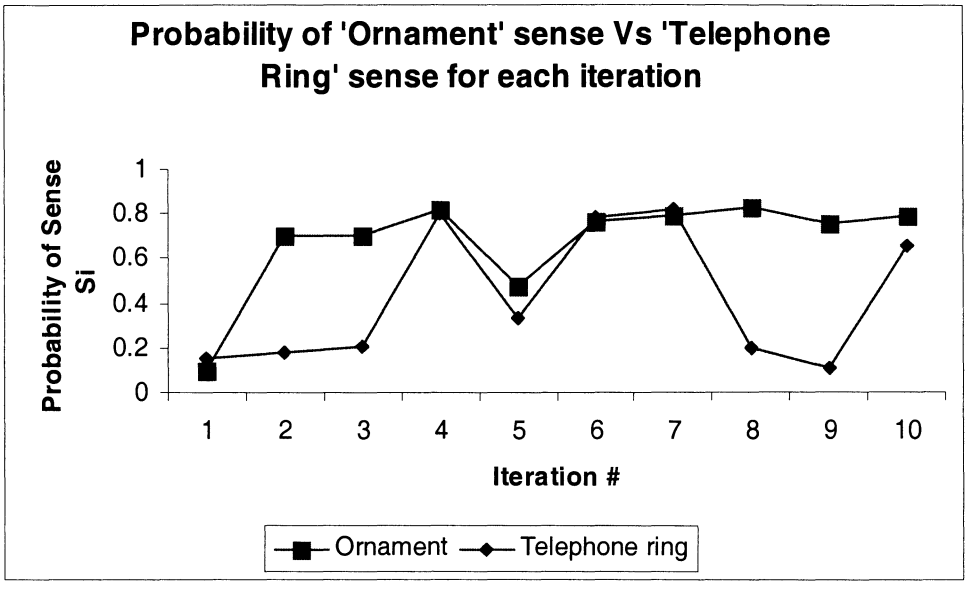


Figure 22) Word : ring



References

Belmore, S.M. ; Yates, J.M. ; Bellack, D.R. ; Jones, S.M. and Rosenquist S.E. (1982) "*Drawing inferences from concrete and abstract sentences.*" Journal of Verbal Learning and Verbal Behaviour, 21, 338-351.

Berger, Adam. (2001) "*Statistical machine learning for information retrieval*", April 2001, CMU-CS-01-110, 20.

Bruce, R. and Wiebe, Janyce. (1994) "*Word Sense Disambiguation using Decomposable Models.*", 141-142

Cohn, Trevor. (2001) "*Word Sense Disambiguation, a brief survey.*", Computing Machinery and Intelligence(Mind, Vol. 59, No. 236, pp. 433-460)

Gorman, A. M. (1961). "Recognition memory for nouns as a function of abstractness and frequency. Journal of Experimental Psychology", 61, 23-29.

Greene, B and Rubin, G. (1971). "Automatic grammatical tagging of English - Technical report.", Department of English, Brown University, 20-23.

Hirst G. (1987). "*Semantic Interpretation and the Resolution of ambiguity*", Cambridge University Press.

Holmes, V. M. and J. Langford (1976). "*Comprehension and recall of concrete and abstract sentences.*" Journal of Verbal Learning and Verbal Behaviour 15, 559-566.

Honavar, V., Parekh, R. and Yang, J. (1999a). *Machine Learning. Invited article. In: Encyclopedia of Electrical and Electronics Engineering*, Webster, J. (Ed.), New York: Wiley.

http://dictionary.reference.com/search?q=expert%20system page retrieved on may 30 2004

http://dictionary.reference.com/search?q=knowledge%20 retrieved on August 17 2003

Ide, N. and Veronis, J. (1998). "*Word Sense Disambiguation: The State of the Art.*", Computational Linguistuics 1998 24(1).

Kaplan, A (1950). "An experimental study of ambiguity in context". Cited in Mechanical Translation, 1, 1-3.

Karov, Y and Edelman, S. (1996). "Learning similarity-based word sense disambiguation from sparse

data." In E. Ejerhed and I. Dagan, editors, Proceedings of the Fourth Workshop on Very Large Corpora, Copenhagen.

Klee, H. and M. W. Eysenck (1973). "*Comprehension of concrete and abstract sentences. Journal of Verbal Learning and Verbal Behaviour",* 12, 522-529.

Krovetz, Robert. et. al(1996). "*Sense Linking in a Machine Readable Dictionary.*", 34-37.

Langley, P. (1995). "Elements of Machine Learning. Palo Alto, CA: Morgan Kaufmann.", 10-14.

Lesk. (1986). "*Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone.*" In the proceedings of SIGDOC '86 Conference, ACM, 24-26.

Madhu, Swaminathan and Dean W. Lytle. (1965). A figure of merit technique for the resolution of non-grammatical ambiguity. Mechanical translation, 8(2):9--13.

Matthews, P. H. (1974). "*Morphology.*" Cambridge University Press, London.

Miller et al.(G.A. Miller; R. Beckwith; C. Fellbaum; D. Gross and K. J. Miller). (1990). "*Introduction to*

WordNet: An on-line lexical database.",
International Journal of Lexicography, 3(4).

Miller, George. A ; Richard Beckwith ; Christiane Fellbaum ; Derek Gross ; K. Miller and Randee Tengi. (1993)."*Five papers on WordNet",* Princeton University. 3(4) 235-312.

Pustejovsky, J. (1995). "*The Generative Lexicon. Cambridge MA, MIT Press."*

Schutze, (1992). "*Dimensions of meaning.",* Proceedings of Supercomputing 1992, 787-796.

Small, (1980). "*Word Expert Parsing: a theory of distributed word-based natural language understanding.",* 9-13.

Stevenson, Mark. (2002). "*The case of combinations of knowledge sources",* 67-79.

Turing, A.M. (1950). Computing machinery and intelligence. Mind, 59, 433-460.

Veronis J. and N. Ide. (1990). "*Word Sense Disambiguation with very large neural networks extracted from Machine Readable Dictionaries.",* Proceedings of the 13th international conference on computational linguistics, 2-4.

Veronis, Jean and Nancy Ide. (1990). *"Word Sense Disambiguation with Very Large Neural Networks Extracted from Machine Readable Dictionaries,"* in Proceedings COLING-90, 389--394.

Weaver, W. (1949). *"The mathematics of commmunication"*, 11-15.

Wilks. (1972). *"Grammar, meaning and machine analysis of language."* London and Boston: Routledge.

Wilks . (1978). Making preferences more active (Artificial Intelligence), 40-42

Wilks, Yorick and Stevenson, Mark. (1996). *"The grammar of sense: Is word sense tagging much more than part-of-speech tagging?", Technical Report CS-96-05*, University of Sheffield, Sheffield, United Kingdom. 5-7.

Yarowsky D. (1994). "One sense per collocation.", In ARPA Human Language Technology Workshop, 266-271.

Yarowsky, David. (1994a). *"Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French."* Proceedings of the 32nd Annual Meeting of the Association for

Computational Linguistics, Las Cruces, New Mexico, 88-95.

Yarowsky, D. (1994b). "*A comparison of corpus-based techniques for restoring accents in Spanish and French text.*" Proceedings of the 2nd Annual Workshop on Very Large Text Corpora. Las Cruces, 19-32.

Yarowsky, David. (1995). "*Unsupervised word sense disambiguation rivaling supervised methods.*" Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, June 1995, Cambridge, Massachusetts, 189-196.

Yngve, V.H. (1955). "*The translation of languages by machine.*", In E.C. Cherry (Ed.), Infomation theory, third London symposium. London: Butterworths.