UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of master's thesis by

SIDDHARTH PATWARDHAN

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Dr. Ted Pedersen

Name of Faculty Adviser

Signature of Faculty Advisor

Date

GRADUATE SCHOOL

Incorporating Dictionary and Corpus Information into a Context Vector Measure of Semantic Relatedness

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Siddharth Patwardhan

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

August 2003

Acknowledgments

I would like to take this opportunity to thank a number of people, without whose support and encouragement this thesis would not have been possible.

Firstly, I would like to thank my advisor Dr. Pedersen for being so thorough and patient, and for seeing me through this research till the end.

I would like to thank my committee members, Dr. Gallian and Dr. Turner, for going over the thesis so carefully and for their insightful suggestions.

I would also like to thank Bano, whose work we built upon and who was full of ideas throughout.

I thank my fellow NLP group members – Saif, Bridget and Amruta – for their ideas and suggestions and my colleague Navdeep for proof-reading and providing her thoughts on the thesis.

I am grateful to Jason Rennie for providing a wonderful interface to WordNet and to Mona Diab for her feedback on the measures. I am also grateful to Diana Inkpen for her insights on the Vector measure.

Finally, I would also like to thank Linda, Lori, Jim, and the faculty of the Computer Science Department at UMD for their valuable support and encouragement the past two years.

Contents

List of Figures

List of Tables

Abstract

Humans are able to judge the relatedness of words (concepts) relatively easily, and are often in general agreement as to how related two words are. For example, few would disagree that “pencil” is more related to “paper” than it is to “boat”. Miller and Charles (1991) attribute this human perception of relatedness to the overlap of contextual representations of words in the human mind, and there is at least some understanding of how humans are able to perform this task. However, it remains an open question as to how to create automatic computational methods that assign relatedness values or scores to pairs of concepts. A number of measures of relatedness have been proposed, most of them relying on information taken from the lexical database WordNet, and possibly augmented with corpus based statistics.

In this thesis we study a number of such measures, and offer various refinements to those proposed by Resnik (1995), Jiang and Conrath (1997) and Lin (1998). We then compare these measures along with three others in the context of a human relatedness study and in word sense disambiguation experiments. We find that the measures of Jiang and Conrath (1997) and Banerjee and Pedersen (2003) offer various advantages. With these results in mind, we propose and evaluate a new measure based on context vectors that combines the content of dictionary definitions with statistical information derived from large corpora. This measure is unusually flexible and robust, in that it does not depend on the structure of any particular dictionary, and it can incorporate information derived from any given corpus of text.

1 Introduction

Semantic relatedness refers to the degree to which two concepts are related (or not). Humans are able to easily judge if a pair of concepts are related in some way. For example, most would agree that *paper* and *pencil* are more related than are *car* and *toothbrush*. This thesis examines the question of how semantic relatedness can be approached from a computational point of view, and results in a new measure of relatedness that will be shown to be both effective and adaptable.

There is, no doubt, a deep psychological explanation behind human perceptions of relatedness. While the exact nature of this remains a fascinating question, in this thesis we consider semantic relatedness from a more practical point of view. We try to observe how humans use this notion in their everyday lives. Knowing what concepts are related (or not) may be considered a part of a human's *common sense*, which is used in nearly every aspect of human thought and action. For example, consider the following sentence: *My son loves baseball, so I got him a bat and glove.* A combination of common sense and domain knowledge about sports makes it clear that the *bat* being referred to is one that is used to hit balls, and is not associated with the well known mammal. These are the kinds of problems that humans solve quickly and without a great deal of conscious thought, based on a combination of their real world knowledge and common sense.

The reader may well wonder that if it were possible to develop computer programs made the same kinds of determinations, could we someday talk to computers, the way we do with humans? No one can answer that just yet, but a reasonable question might be the following: *can we automate and quantify semantic relatedness, so as to correspond with human judgment?* The answer to this question, based on previous research and this thesis, is a qualified yes.

However, this is a challenging problem. There are a wide range of different ways that concepts can be related, and it may require a certain amount of specialized knowledge to realize such cases. For example, on first reflection the automotive sense of *tire* and the shade-giving type of *tree* may not seem to be related. However, if one is aware that tires are made of rubber, and rubber comes from trees, then they may be more related than first realized. In addition, humans are not always in complete agreement on relatedness judgments, since these can be affected by uniquely personal experiences. For example, a particular person may consider *tree* and *car* to be highly related because that person parks her car under a big tree everyday.

Despite the caveats issued above, it is still reasonable to say that humans are largely in agreement on the

semantic relatedness of concepts. This has been verified by several repetitions of human studies across a number of years, and we utilize the results of such experiments when evaluating our computational work.

Before proceeding, we must clarify the relationship between *words*, *concepts* and *word senses*. Concepts are real world objects, notions or ideas that are represented in text or speech by words. For example, the concept of a stone would be represented by the word *stone*. In addition, it may well be represented by the word *rock* or *pebble*. Hence, the same concept may be represented by different words. Also, the concept need not be a solid object. It could be an abstract thing, like art, or an action, like walking. Each such concept has a number of words that represent it. Not only that, but a single word may represent a number of concepts. For example, the word *bank* could mean the financial institution concept or the river bank concept. The different meanings of a word are known as word senses. A word could, therefore, correspond to a number of concepts, while a word sense corresponds only to a single concept. Due to this equivalence of word senses and concepts, in this thesis we use the terms *concepts* and *word sense* interchangeably.

The reader may have noted that in our previous examples, dealing with semantic relatedness and human perception of relatedness, we have been showing *words* in the text but then referring to the relatedness of *concepts* or of *word senses*. This is simply a convenience. In general, measures of relatedness focus on underlying concepts or word senses, and when a pair of words are presented in work such as this, it is presumed that the obvious senses are intended (unless otherwise indicated). For example, if we present the example pair *bank* and *money*, we would presume that we are referring to the most obvious senses, which are likely the financial ones.

Concepts may be related by virtue of being similar in appearance or function (e.g., *chair* and *stool*), being opposites (e.g., *rich* and *poor*), or having some other defined relationship between them, such as one concept being a *a type of* or *a part of* the other. For example, a *penguin* is a type of *bird*, and a *beak* is a part of a *penguin*. However, two concepts can be related without there being a defined relationship between the words. They may be considered highly related if they just happen to occur together very often in everyday use. For example *bargain* and *book* may be considered related, simply because the collocation *bargain book* is relatively common.

A number of measures of semantic relatedness have been developed and evaluated by different researchers, and this thesis will review and discuss some of the most significant of these. In addition, this thesis will show several refinements to these existing measures, present a new task oriented measure of evaluation, and

conclude with the presentation of a new measure of semantic relatedness.

However, before we go into the gory details of what we did and how we did it, we would like to explore a bit further the question of *why* we have worked on this problem. Apart from the importance of semantic relatedness in cognitive sciences and psychology, it has a number of significant applications in Natural Language Processing. For example, semantic relatedness of words can be effectively used for query expansion in improving information retrieval. Suppose a user has requested information about *New York City rental property*. If the system knows that *apartments* are a form of rental property, then the query can be modified such that this additional terminology is employed. Document clustering based on content also provides a direct application. Here a measure of the semantic relatedness of the content of different documents could be used to cluster them into semantically related groups.

In this thesis we will review various measures that are based on the lexical database *WordNet* [8]. Some measures extend WordNet's content with statistical information derived from large corpora, while others employ different aspects of WordNet's structure and content.

With various measures of semantic relatedness available, a major challenge has been to find a reasonable basis for their comparison. Budanitsky and Hirst [6] compare the performance of several measures of relatedness with the results of human studies of semantic relatedness that have been previously published.

Due to the relatively small number of human studies that have been conducted (due to the expense and difficulty of arranging such an effort), we refer to the same body of previous work. In particular we rely upon the landmark Rubenstein and Goodenough [25] study from 1965, and the later replication of those first results by Miller and Charles [19] in 1991.

Both Budanitsky and Hirst and this thesis find that automatic computational measures perform relatively well when compared to human judgment. In addition, Budanitsky and Hirst suggest the study of the impact of different measures of relatedness when applied to a real world problem. In their case they chose spelling correction. We follow their lead, and present an novel means of comparing measures of relatedness. We have devised a method of *word sense disambiguation* that can be used as the basis for carrying out an extensive comparison of these measures, and this thesis also presents the results of that study.

Word sense disambiguation is the problem of selecting the most appropriate meaning or sense of a word, based on the context in which it occurs. While this is usually an easy task for a human, it is a challenging

problem for a computer program since a machine does not have the benefit of a lifetime of experience nor does it have a deep knowledge of language. We evaluated different measures of relatedness via an adaptation of the famous Lesk [15] algorithm for word sense disambiguation.

This is a method that uses dictionary definitions (or glosses) of surrounding words to determine the correct sense of a particular word. From this point forward we will refer to the word to be disambiguated as the *target word*, and the surrounding words in the text as the *context*. According to the Lesk Algorithm the sense of the target word whose dictionary definition has the maximum overlap of words with definitions of senses of other words in the context is the sense that is selected as the intended sense of the target word.

The basic hypothesis underlying Lesk's method for word sense disambiguation is that the sense of the target word should be related to the other words in the context, and that the degree of relatedness can be captured via a measure based on gloss overlaps. Banerjee and Pedersen [3] extend Lesk's algorithm to extend the dictionary definitions with additional definitions from concepts found in the rich network of word sense relations in WordNet. This thesis demonstrates that the Lesk framework does not depend on using gloss overlaps as a measure of relatedness, and in fact any measure of relatedness can be used to carry out disambiguation. It was this observation that allows us to carry out the comparative study of different measures of relatedness as applied to disambiguation of the SENSEVAL-2 data, which will be discussed in more detail later.

However, by way of summary we compared the following six measures of relatedness to see how well they fared in comparison with human relatedness perception and with respect to word sense disambiguation:

- the Resnik measure [24],
- the Lin measure [16],
- the Jiang-Conrath measure [12],
- the Leacock-Chodorow measure [14],
- the Hirst-St.Onge measure [10], and
- extended gloss overlaps [4].

(The first five measures mentioned above were the object of the study by Budanitsky and Hirst, mentioned earlier in this section.)

We found that Banerjee and Pedersen's measure of extended gloss overlaps fared well in both evaluations. However, we have some reservations about measures based strictly on the contents of a dictionary. Glosses or definitions of words are not meant to be complete descriptions of the concepts represented by the words. They contain a minimal description of the different senses of a word. Therefore, measuring relatedness by counting the word overlaps in the dictionary definitions is highly sensitive to the size of the definitions and to the dictionary used.

We believe that gloss based measures of relatedness could be improved upon by augmenting the definitions with data derived from large corpora of text. This belief is supported by the observation that the other measure that fared well in our comparative studies was that of Jiang and Conrath [12]. This measure uses statistics from a large corpus in the form of *information content* of word senses in addition to taking advantage of the structure of the *is-a* hierarchy in WordNet.

Due to the success of both extended gloss overlaps and the measure of Jiang and Conrath, this thesis creates a new measure of semantic relatedness that represents each word sense as a multidimensional vector of word frequencies. We build the multidimensional vectors based on the notion of *context vectors* described by Schütze [26]. These vectors combine dictionary definitions of the word senses with co-occurrence data from a large corpus. Semantic relatedness is then measured simply as the nearness of the two vectors in the multidimensional space (the cosine of two normalized vectors). One of the strengths of this measure is that, although our implementation of the measure is tied with WordNet, the basic idea of the Vector measure can be used with any dictionary. Also, this measure is not restricted by any particular part of speech and can find the relatedness between concepts from any part of speech.

After developing our measure we re-did the previously mentioned comparative experiments. We compared the six measures with respect to the human perception of semantic relatedness and within a word sense disambiguation task. Our results show that the Extended Gloss Overlaps measure and the Vector measure correspond very closely to the human perception of semantic relatedness. We found that the new Vector measure did not fare quite as well in the word sense disambiguation task, but we believe that there are reasonable explanations for that and are optimistic that these results will improve.

To summarize briefly, this thesis has resulted in the following contributions to computational measures of semantic relatedness.

1. In cooperation with Banerjee, we cast the extended gloss overlap technique as a measure of relatedness.

2. We extended the Adapted Lesk Algorithm of Banerjee and Pedersen such that any measure of relatedness can be used to carry out disambiguation.

3. We carried out an extensive and novel evaluation that assessed the effectiveness of six different measures of relatedness when applied to word sense disambiguation.

4. We refined the measures of Resnik, Jiang–Conrath, Lin and Hirst and St. Onge in order to include them in the just mentioned comparative study.

5. Based on the results of our comparative studies, we created a new context vector based measure that combines corpus data with dictionary definitions. The new measure is independent of any part of speech restrictions and can be implemented independent of the dictionary used.

6. We conducted a second set of comparative experiments relative to human relatedness and word sense disambiguation in order to evaluate the Vector measure. We find that the Vector measure performs exceptionally well relative to human relatedness, and reasonably well at word sense disambiguation.

7. We have released (via the CPAN archive) a freely available software package that implements all of the measures of relatedness discussed here.

2 Semantic Relatedness

Measuring the semantic relatedness of concepts is an intriguing problem in Natural Language Processing. Various approaches that attempt to approximate human judgment of relatedness, have been tried by researchers. In this section we look at a few WordNet–based measures of semantic relatedness that we propose to compare. Approaches to measuring semantic relatedness that we have not experimented with in this thesis are discussed in the section on related work (section 7).

It is important to note that, although *semantic similarity* and *semantic relatedness* are sometimes used interchangeably, the term *relatedness* is more general than *similarity*. Budanitsky and Hirst [6] discuss this point and they say that similarity usually refers to concepts that are related because they look alike. For example, *table* is similar to *desk*. On the other hand, dissimilar concepts like *wheel* and *spoke* may be semantically related. In this thesis, we deal with measures of semantic relatedness.

Before we delve into the intricacies of measuring semantic relatedness, a quick introduction to WordNet is in order, since all the measures described here are based on WordNet.

2.1 WordNet

The creators of WordNet refer to it as an electronic lexical database [8]. This is a convenient but over–simplified description of a very complex resource. WordNet can be visualized as a large graph or semantic network, where each node of the network represents a real world concept. For example, the concept could be an object like a *house*, or an entity like a *teacher*, or an abstract concept like *art*, and so on.

Every node consists of a set of words, each representing the real world concept associated with that node. Thus, each node is essentially a set of synonyms that represent the same concept. For example, the concept of a *car* may be represented by the set of words $\{$*car, auto, automobile, motorcar*$\}$. Such a set, in WordNet terminology, is known as a *synset*. A synset also has associated with it a short definition or description of the real world concept known as a *gloss*. The synsets and the glosses in WordNet are comparable to the content of an ordinary dictionary.

What sets WordNet apart is the presence of links between the synsets – the edges of the graph mentioned above. Each link or edge describes a relationship between the real world concepts represented by the synsets

Table 1: Relations between synsets defined in WordNet

Relation	Description	Example
Hypernym	is a generalization of	*furniture* is a hypernym of *chair*
Hyponym	is a kind of	*chair* is a hyponym of *furniture*
Troponym	is a way to	*amble* is a troponym of *walk*
Meronym	is part/substance/member of	*wheel* is a (part) meronym of a *bicycle*
Holonym	contains part	*bicycle* is a holonym of a *wheel*
Antonym	opposite of	*ascend* is an antonym of *descend*
Attribute	attribute of	*heavy* is an attribute of *weight*
Entailment	entails	*ploughing* entails *digging*
Cause	cause to	to *offend* causes to *resent*
Also see	related verb	to *lodge* is related to *reside*
Similar to	similar to	*dead* is similar to *assassinated*
Participle of	is participle of	*stored* (adj) is the participle of "to *store*"
Pertainym of	pertains to	*radial* pertains to *radius*

that are linked. For example, relationships of the form "a vehicle *is a kind* of conveyance" or "a spoke *is a part of* a wheel" are defined. Other relationships include *is opposite of, is a member of, causes, pertains to,* etc. Table 1 shows the list of relations defined in WordNet. The network of relations between word senses present in WordNet encodes a vast amount of human knowledge. This gives rise to a great number of possibilities in the way it could be used for various Natural Language Processing (and other) tasks.

Figure 1 focuses on a small portion of the structure of WordNet and illustrates the nodes and edges of the semantic network just described.

The synsets in WordNet are divided into four distinct categories, each corresponding to four of the parts of speech – nouns, verbs, adjectives and adverbs. Most of the relationships defined between the synsets are restricted to a particular part of speech and do not cross part of speech boundaries. Exceptions are the *pertains to* and *attribute* relationships that exist between adjectives and nouns. Thus, the set of relations defined on the synsets in WordNet, divide them into four almost disjoint regions.

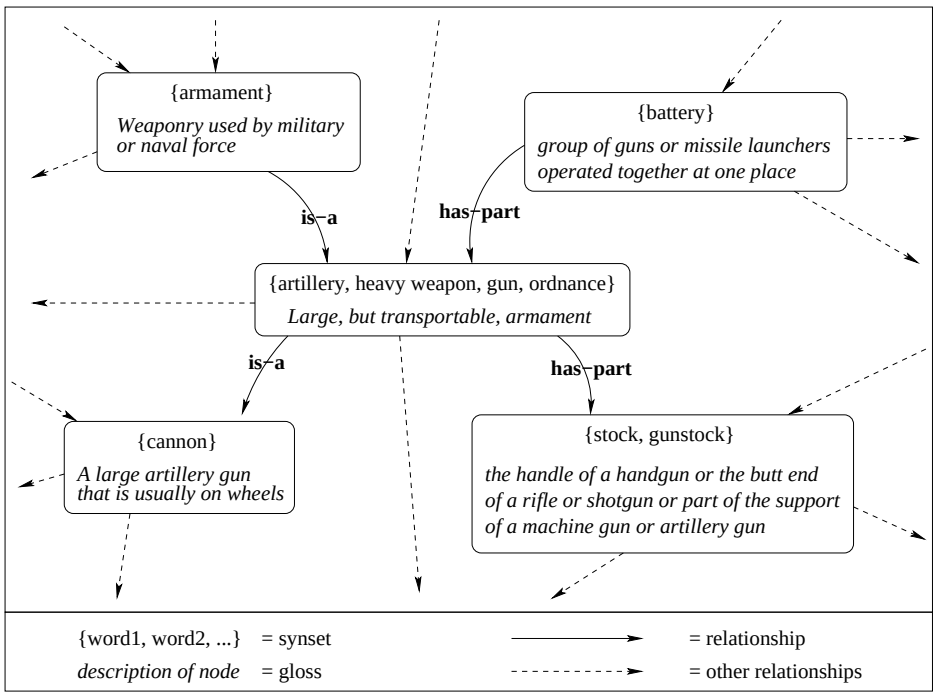

Figure 1: An illustration of *synsets* and *relations* in WordNet

One of the relations in WordNet of interest to us, mainly because of its structure and utility in measuring semantic relatedness, is the *is a kind of* relationship or simply *is a*. This relationship between synsets is restricted to nouns and to verbs. This relation organizes the noun and verb synsets into large hierarchies or trees. Each tree has a single root node. The more general concept nodes are ancestors of more specific concept nodes. We say that the more general concepts *subsume* the more specific concepts. For instance, *entity* is the most general concept in one of the noun hierarchies and is the root node of the tree. It subsumes other more specific concepts such as *furniture*, *bicycle*, etc, which are lower down in the tree. Similarly, *furniture* may subsume other concepts such as those of *chair* or *table*. There exist 9 such hierarchies in the WordNet nouns, while there are about 628 hierarchies for verbs. The large number of hierarchies in verbs is due to the fact that the verb hierarchies are, on average, much shorter and broader than the noun hierarchies. The average depth of the noun hierarchies is about 12.5 nodes, while that of the verb hierarchies is about 2.3 nodes. Each of the verb hierarchies, therefore, covers a much smaller portion of the synsets, as compared to the noun hierarchies. This makes the verb hierarchies a lot less effective in the relatedness measures that we describe later in the section. Figure 2 shows an example of the *is-a* hierarchy in WordNet.

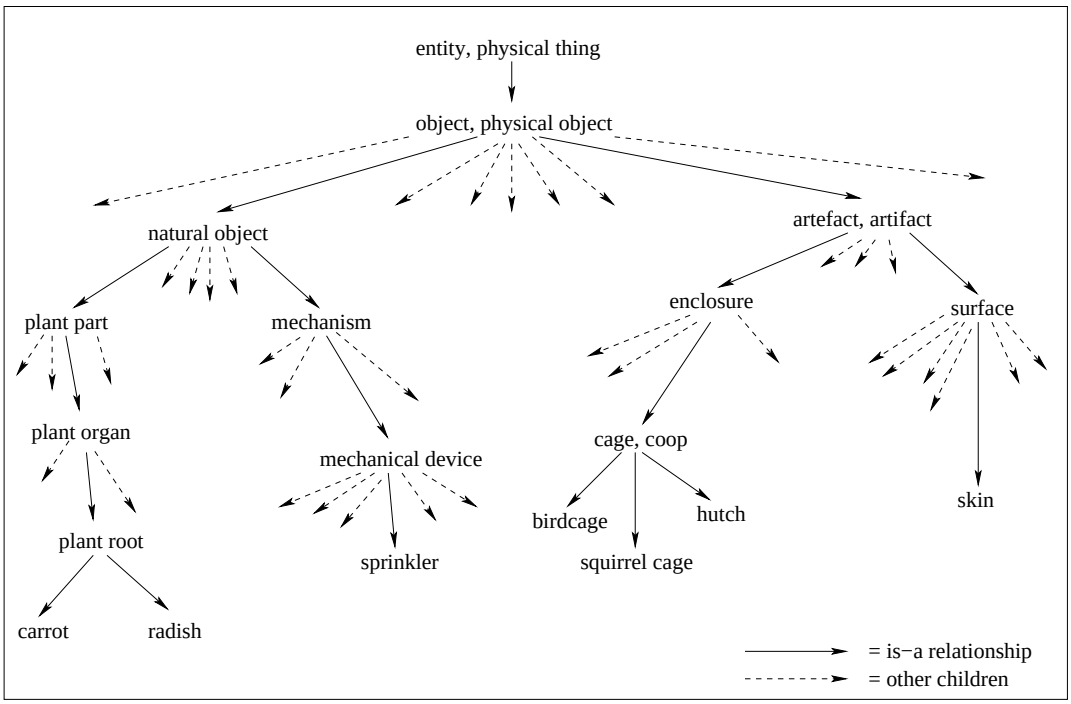

Figure 2: A schematic of the *is-a* hierarchy in WordNet

For all our experiments we used version 1.7.1 of WordNet.

2.2 Measuring Semantic Relatedness

Given the vast store of human knowledge encoded in WordNet, it has been used by many researchers in developing measures of semantic relatedness. Some use only the structure and content of WordNet to measure semantic relatedness. Other approaches combine statistical data from large corpora with the structure of WordNet to give us a score of semantic relatedness.

2.2.1 The Leacock-Chodorow Measure

An intuitive method to measure the semantic relatedness of word senses using WordNet, given its tree-like structure, would be to count up the number of links between the two synsets. The shorter the length of the path between them, the more related they are considered. Such a measure had been experimented with

by Rada et al [23] for measuring semantic relatedness of medical terms, using a medical taxonomy called MeSH. Their measure performed rather well. A measure suggested by Leacock and Chodorow [14] does almost this, using WordNet. The measure suggested by Leacock and Chodorow considers only the *is a* hierarchies of nouns in WordNet. Since only noun hierarchies are considered, this measure is restricted to finding relatedness between noun concepts. The noun hierarchies are all combined into a single hierarchy by imagining a single root node that subsumes all the noun hierarchies. This ensures that there exists a path between every pair of noun synsets in this single tree. To determine the semantic relatedness of two synsets, the shortest path between the two in the taxonomy is determined and is scaled by the depth of the taxonomy. The following formula is used to compute semantic relatedness:

$$related_{lch}(c_1, c_2) = -log\left(\frac{shortestpath(c_1, c_2)}{2 \cdot D}\right) \tag{1}$$

where c_1 and c_2 represent the two concepts, $shortestpath(c_1, c_2)$ specifies the length of the shortest path between the two synsets c_1 and c_2, and D is the maximum depth of the taxonomy. For WordNet 1.7.1, the value of D turns out to be 19.

This method assumes the size or weight of every link in the taxonomy to be equal. This is a false assumption. It is observed that lower down in the hierarchy, concepts that are a single link away are more related than such pairs higher up in the hierarchy. This simple approach, however, does relatively well, despite its lack of complexity.

Some highly related approaches attempt to overcome this disadvantage of simple edge counting by augmenting the information present in WordNet with statistical information from large corpora.

2.2.2 The Resnik Measure

Statistical information from large corpora is used to estimate the *information content* of concepts. The idea of information content was introduced by Resnik [24], in his paper that describes a novel method to compute semantic relatedness.

In brief, information content of a concept measures the specificity or the generality of that concept, i.e. how specific to a topic the concept is. For example, a concept like *sprinkler* is a highly topical concept and would have high information content. On the other hand, a more general concept such as *artifact* would have a much lower information content.

Computation of information content is based on the subsuming property of the *is a* hierarchy. Suppose we come across concept c_1 in a discourse, which is subsumed by concept c_2 in the *is a* hierarchy of WordNet. Then the occurrence of concept c_1 in the text implies the occurrence of c_2 in the text, which is explained by the fact that c_1 is a kind of c_2. For example, suppose we come across the concept *chair* in a given text. Because *chair* is a kind of *furniture*, we can always say that text contains the concept of *furniture*. And further, without being wrong, we can say that it speaks of an *object*. Thus, *chair*, *furniture* and *object* all represent the same concept in the text at varying degrees of specificity.

To find information content we first compute the frequency of occurrence of every concept in a large corpus of text. Every occurrence of a concept in the corpus adds to the frequency of the concept and to the frequency of every concept subsuming the concept encountered. We note that by this process the root node includes the frequency count of every concept in the taxonomy. It is incremented for every occurrence of every concept in the taxonomy.

Counting of concepts from a corpus is, however, not as trivial as described. The inherent ambiguity of words poses the problem in determining the occurrence of concepts in the corpus. Unless we have a sense-tagged corpus, we will not be able to tell if the occurrence of the word *bank* in the corpus refers to the financial-institution sense of bank or to the river-bank sense of bank or to some other sense of bank. In other words, each word sense refers to a unique concept, and words can have multiple senses.

Resnik overcomes the problem of ambiguity by distributing the count of a word over all senses of the word. Thus, if the word *bank* is encountered 50 times in the text and *bank* has 10 senses in WordNet, then each of these 10 concepts would receive a count of 5. This assumes an equal distribution of the senses in text.

In this thesis we introduce a different method of counting concept frequencies and compare the effect of our counting to that of the Resnik counting. In our method of counting, rather than distributing the frequency count of a word across its senses, we assign that count itself to all the senses. In the preceding example, using our counting, each of the 10 senses of *bank* would receive a count of 50 instead.

The primary reason for using a different counting scheme from the one described by Resnik is that in the Resnik counting we observe that by distributing the word counts across the senses of the word, we assign higher relative frequencies to words having fewer senses.

Information content is defined as the negative logarithm of the probability of occurrence of the concept in a

large corpus. Thus, with these frequency counts computed from a large corpus, we arrive at the following formula for information content:

$$IC(c) = -log\left(\frac{freq(c)}{freq(root)}\right) \tag{2}$$

where $IC(c)$ is the information content of concept c, $root$ is the root node of the taxonomy and $freq(c)$ and $freq(root)$ are the frequency counts of these concepts.

Another issue that had to be addressed was that of 0 frequency counts. If c had a frequency count of 0 in the above formula, we would end up with an undefined information content value. We handle this in two ways. The first is to allow information content to have a value of 0 and then have the measures have special handling for an information content value of 0. The second was to smooth the frequency counts. We use add-1 smoothing and compare the effects of the two methods. Smoothing is a way of assigning small non-zero frequency values to concepts not observed in a corpus of text. In add-1 smoothing, a value of 1 is added to the frequency of occurrence of each of the concepts. This causes the concepts not observed in a corpus to have a small non-zero frequency.

Resnik defines the semantic relatedness of two concepts as the amount of information they share in common. He goes on to elaborate that the quantity of information common to two concepts is equal to the information content of their *lowest common subsumer* – the lowest node in the hierarchy that subsumes both concepts. For example, in figure 2 the lowest common subsumer of *carrot* and *radish* is *plant root*, while that of *carrot* and *birdcage* is *object*.

$$related_{res}(c_1, c_2) = IC(lcs(c_1, c_2)) \tag{3}$$

where IC determines the information content of a concept and $lcs(c_1, c_2)$ finds the lowest common subsuming concept of concepts c_1 and c_2.

The Resnik measure depends completely upon the information content of the lowest common subsumer of the two concepts whose relatedness we wish to measure. It takes no account of the concepts themselves. This leads to somewhat "coarser" relatedness values. For example, the concept pair *car* and *bicycle* will have the same measure of semantic relatedness as the pair *car* and *all terrain bicycle* because both pairs of concepts have the same lowest common subsumer.

This measure has a lower bound of 0 and no upper bound.

2.2.3 The Jiang-Conrath Measure

A measure introduced by Jiang and Conrath [12] addresses the the limitations of the Resnik measure. It incorporates the information content of the two concepts, along with that of their lowest common subsumer. The measure is a *distance* measure that specifies the extent of *unrelatedness* of two concepts. It combines features of simple edge counting with those of information content introduced in the Resnik measure. The Jiang–Conrath measure is given by the formula:

$$distance_{jcn}(c_1, c_2) = IC(c_1) + IC(c_2) - (2 \cdot IC(lcs(c_1, c_2))) \tag{4}$$

where IC determines the information content of a concept and lcs determines the lowest common subsuming concept of two given concepts.

For the purpose of our experiments, to maintain a scale where a value of 0 indicates unrelated concepts, we invert the value to make it a measure of semantic relatedness.

$$related_{jcn}(c_1, c_2) = \frac{1}{distance_{jcn}(c_1, c_2)} \tag{5}$$

The relatedness would be undefined if there was a 0 in the denominator, which can happen in two special cases:

1. The first case:

   $$IC(c_1) = IC(c_2) = IC(lcs(c_1, c_2)) = 0 \tag{6}$$

   $IC(lcs(c_1, c_2))$ can be 0 if the lowest common subsumer turns out to be the root node, since the information content of the root node is zero. $IC(c_1)$ and $IC(c_2)$ would be 0 only if the two concepts have a 0 frequency count, in which case, for lack of data, the measure returns a relatedness of 0. c_1 and c_2 can never be the root node, since the root node is a virtual node created by us and doesn't really exist in WordNet.

   Thus, in this case we return a relatedness score of 0, indicating insufficient data to assess the relatedness of c_1 and c_2.

2. The second case in which we may have to handle a 0 in the denominator is when

   $$IC(c_1) + IC(c_2) = 2 \cdot IC(lcs(c_1, c_2)) \tag{7}$$

which is more likely to occur in the special case

$$IC(c_1) = IC(c_2) = IC(lcs(c_1, c_2)) \tag{8}$$

This usually happens when c_1, c_2 and $lcs(c_1, c_2)$ turn out to be the same concept.

Intuitively this is the case of maximum relatedness (zero distance), and simply returning a relatedness score of 0, indicating unrelated concepts, would not be right. A more reasonable option is to return an arbitrarily high value, signifying maximum relatedness. But the difficulty is of selecting such a value.

In this thesis, this case is handled by finding the smallest $distance_{jcn}$ greater than 0. This value indicates the maximal relatedness or minimal non-zero distance. To find this value of $distance_{jcn}$, consider equation (4). Here the value of $distance_{jcn}$ is 0 when we have condition specified in equation (8). Now, consider the case that $IC(c_2) = IC(lcs(c_1, c_2))$, but $IC(c_1)$ is just slightly greater than $IC(c_2)$. We want to find the value of distance corresponding to such a case and this would be the value of distance just above 0. From equation (2) we have

$$IC(c_1) = -log\left(\frac{freq(c_1)}{freq(root)}\right) \tag{9}$$

For $IC(c_1)$ to be just slightly more than $IC(c_2)$, we reduce $freq(c_1)$ in the above formula (equation (9)) by 1. Suppose, f is the original frequency of c_1 (and of c_2 and $lcs(c_1, c_2)$), then with the reduced frequency, $IC(c_1)$ becomes

$$IC(c_1) = -log((f-1)/f_{root}) \tag{10}$$

and we have

$$IC(c_2) = IC(lcs(c_1, c_2)) = -log(f/f_{root}) \tag{11}$$

Since frequency is counted in integers, this is the closest $IC(c_1)$ could be to $IC(c_2)$. We then have

$$\begin{aligned}
distance_{jcn} &= IC(c_1) + IC(c_2) - (2 \cdot IC(lcs(c_1, c_2))) && (12)\\
&= IC(c_1) + IC(c_2) - (2 \cdot IC(c_2)) \quad \ldots \text{ since } IC(c_2) = IC(lcs(c_1, c_2)) && (13)\\
&= IC(c_1) - IC(c_2) && (14)\\
&= -log((f-1)/f_{root}) + log(f/f_{root}) && (15)
\end{aligned}$$

Now, suppose we let c_1 and c_2 be the root node, for this computation.

$$\begin{aligned} distance_{jcn} &= -log((f_{root} - 1)/f_{root}) + log(f_{root}/f_{root}) && (16) \\ &= -log((f_{root} - 1)/f_{root}) + log(1) && (17) \\ &= -log((f_{root} - 1)/f_{root}) && (18) \end{aligned}$$

Equation (18) specifies the value of distance that is almost equal to zero. We use this value of "almost zero distance" in the equation for relatedness (equation (5)).

This measure works only with WordNet nouns, has a lower bound of 0 and no upper bound. But we have created an artificial upper bound on the measure in this thesis.

2.2.4 The Lin Measure

Another measure, based on information content of concepts, is described by Lin [16]. The measure is given by:

$$related_{lin}(c_1, c_2) = \frac{2 \cdot IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} \quad (19)$$

For this measure, we have special handling for 0 information content values, since a 0 in the denominator in the above formula would give us an undefined relatedness value. We simply return a relatedness value of 0 if either of the two concepts have an information content of 0. This is because, neither c_1 nor c_2 can be the root node. So their having an information content of 0 implies a lack of data (no frequency count for the concept).

This measure has a lower bound of 0 and an upper bound of 1.

The information content based measures described here, though closely related, give surprisingly different results.

2.2.5 The Hirst-St.Onge Measure

Hirst and St.Onge [10] also use the rich content of WordNet to define relatedness between words. Note that this measure, reports the relatedness between words and not between word senses or concepts. In this

thesis, we modify this relation, so that it becomes a measure of relatedness of concepts. Unlike the above measures that considered only the *is a* hierarchy of nouns, the Hirst-St.Onge measure actually considers all the relations defined in WordNet. All links in WordNet are classified as *Upward* (e.g. part-of), *Downward* (e.g. subclass) or *Horizontal* (e.g. opposite-meaning). Further, they also describe three types of relations between words - *extra-strong*, *strong* and *medium-strong*. Any two words are related by one of these types of relations if they conform to certain rules summarized below.

Extra-strong relations are defined between two instances of the same word. Observe that this specifies a relationship between surface forms of words. Since we are dealing with the semantic relatedness of word senses, we do not consider this category of relations in our experiments.

Two words are related by a *strong* relation under the following conditions.

- If the two words belong to the same synset in WordNet. For example, *car* and *automobile*.
- If the two words belong to two synsets connected by a horizontal link in WordNet. For example, two words that are opposite in meaning, such as *hot* and *cold* have a horizontal link between them.
- If one word is a compound word, the second word is part of the compound word and there exists an *is-a* relation between the synset of the first word and that of the second word in WordNet. For example, *school* and *private_school* have such a relationship.

We assign a score of $2 \cdot C$ to an occurrence of this relation. C is a constant used in the formula for the scoring of a *medium-strong* relation. Hirst and St.Onge use 8 as a value for C in their experiments.

A *medium-strong* relation is defined between synsets connected by a path in WordNet that is not too long and has relatively few changes in direction. The *upward*, *downward* and *horizontal* classification of WordNet relations described earlier in this section, indicate the direction of the relations. The weight of any medium strong path is given by

$$Weight = C - Path\ Length - k \times Changes\ in\ direction \tag{20}$$

where C, k are constants. Medium-strong relations have some additional restrictions regarding the direction that the path may follow. The path between two words with the lowest weight is the one always considered. These three types of relations describe the degree of relatedness of words.

In this thesis we use values of $C = 8$ and $k = 1$ in equation (20). These were the values used by Hirst and St.Onge for their experiments. This sets a lower bound of 0 and an upper bound of 16 on the measure.

2.2.6 Extended Gloss Overlaps as a Measure of Relatedness

Lesk [15] defines relatedness in terms of dictionary definition overlaps of concepts. He describes an algorithm that disambiguates words based on the extent of overlaps of their dictionary definitions with those of words in the context. The sense of the target word with the maximum overlaps is selected as the assigned sense of the word. The hypothesis that dictionary definition overlaps can measure semantic relatedness underlies this algorithm.

Banerjee and Pedersen [3] adapt the Word Sense Disambiguation algorithm, described by Lesk [15], to WordNet. Since the Lesk algorithm was algorithm was designed before the creation of WordNet, it was mainly based on traditional dictionaries. Banerjee and Pedersen enhance the Lesk algorithm with the rich source of knowledge present in WordNet. The algorithm proceeds by taking each of the words in the context of the ambiguous word and considers the glosses of all words connected to these by various WordNet relations. The overlap of each of these glosses with glosses of words connected to each sense of the ambiguous word is determined and is used to compute a score for these senses. The sense with the highest score is the selected.

In this thesis, we propose that this adaptation of Banerjee and Pedersen, in fact, can be thought of as a measure of semantic relatedness. It is called the extended gloss overlaps measure. This method scores a pair of glosses by finding the number of word strings that are common to the two extended glosses. Multiple word matches are scored higher than single word matches. This is done by adding the square of the number of consecutive words matched, to the score of the gloss pair. For example, if the string *space shuttle* occurs in two glosses, 4 is added to the score of the gloss pair, for this two word string match.

This measure is called the *extended* gloss overlap measure, because rather than specifying the relatedness as the score of the glosses of the two concepts alone, the glosses of words related to the concepts are taken into consideration. This process is described in a little more detail below.

Consider the first of the two concepts c_1 and a set C_1 of glosses corresponding to c_1 (initially empty). The gloss of concept c_1 is added to the set C_1. A gloss is identified corresponding to each of the WordNet

relations. For a WordNet relation r, a gloss is created by concatenating the glosses of all concepts related to c_1 by relation r. All such glosses (corresponding to all the WordNet relations) are added to set C_1. Similarly we create a set C_2 for the second concept c_2.

To find the relatedness of c_1 and c_2, gloss overlap scores for each gloss in C_1 with each gloss in C_2 are added and the sum is the semantic relatedness of concepts c_1 and c_2. This is precisely what is done by Banerjee and Pedersen in their adaptation of the word sense disambiguation algorithm.

This measure has a lower bound of 0 and no upper bound.

3 Measuring Semantic Relatedness Using Context Vectors

We conducted some preliminary experiments [21] to compare the measures of semantic relatedness described by Resnik [24], Jiang-Conrath [12], Lin [16], Leacock-Chodorow [14], Hirst-St.Onge [10] and Banerjee-Pedersen [4]. On analyzing the results we found that the information content based measure described by Jiang and Conrath and semantic relatedness based on extended gloss overlaps (Banerjee and Pedersen) fared the best in a Word Sense Disambiguation task.

As described in section 2, the Jiang-Conrath measure uses the knowledge from a large corpus in the form of information content of word senses. This knowledge is used in conjunction with the rich network of relationships between word senses provided by WordNet, to assign a quantitative value to the semantic relatedness of word senses.

The extended gloss overlap measure of Banerjee and Pedersen, on the other hand, uses an entirely different technique of measuring the extent of overlap of WordNet definitions of word senses. It is not assisted, in any way, by an additional knowledge source like a large corpus.

Even though both the measures do quite well, we note that the Extended Gloss Overlap measure suffers from the disadvantage that it is dependent on exact matches of words. Thus, the presence of a content word like *spoon* in two glosses would contribute to their overlap score. However, if one of the two glosses contained *spoon* and the other contained *spoons*, the overlap would be missed. Conceptual matches like *spoon* and *silverware* would not even be considered.

In order to overcome this disadvantage of the extended gloss overlaps measure, we consider ways of augmenting the words in the glosses with data from external sources. We use an alternate representation and an alternate matching scheme of WordNet concepts that is not as short and not as exact as a WordNet gloss, but describes the concept in a broader sense. Some work by Schütze [26] and Inkpen and Hirst [11] gives us some direction towards such a representation. Like the Extended Gloss Overlaps measure, it works on all parts of speech and like the Jiang-Conrath measure it takes advantage of corpora.

3.1 Introduction to Context Vectors

Schütze introduces a unique application of *context vectors* in his paper on *Automatic Word Sense Discrimination* [26]. Such multidimensional vectors of word frequencies have been traditionally used in Information Retrieval. Schütze uses them in *Word Sense Discrimination*, which is the process of clustering together passages of text, each of which contain an instance of a particular ambiguous word. Clusters are formed such that the each cluster contains all those passages with the same sense of the target (ambiguous) word. This process is one step short of *Word Sense Disambiguation*, since an actual sense is not assigned to the members of the cluster. In word sense discrimination, we are able to say whether two instances of the target word are used in the same sense, but we are unable to say what they mean.

In order to perform word sense discrimination, Schütze represents passages of text as vectors in a multidimensional space. His algorithm is based on a hypothesis by Miller and Charles [19], that humans determine the similarity in the meanings of words from their contexts. For example, consider the sentences "He filed a *suit* in court" and "He wore the new *suit* to the party". We are easily able to tell from the words preceding and following (i.e. from the context of) the word *suit* in the two sentences that the two instances of the word *suit* have entirely different meanings. In the first instance, the presence of the noun *court* and the verb *file* motivates us to believe that the word *suit* in the sentence alludes to a law-suit. Similarly, the words *wore*, *new* and *party* ascertain that, in the second sentence, the word *suit* speaks of clothing. We could convince ourselves that the the word *suit* in the first sentence would be closely related to a word like *judge*, since both are more likely to have similar words (i.e. legal terms) around them in sentences. This supports the hypothesis by Miller and Charles of the similarity in meanings of words being determined by their contexts.

Schütze uses this notion by representing the context of the words using vectors mapped into a multidimensional space. The dimensions of this space are defined by the number of words present in a "word space". The word space is just a list of words used to form the vectors. The words in the word space are selected either by using a frequency cut-off or by using the χ^2 test of association on a corpus of text. In order to represent the multidimensional space *word vectors*, *context vectors* and *sense vectors* are introduced. Every word in the word space has a corresponding word vector. The word vector corresponding to a given word is calculated as a vector of integers. The integers are the frequencies of occurrence of each word from the word space in the context of the given word in a large corpus. Thus, each word in the word space represents a dimension of the vector space.

Once the word vectors for all words of the word space are calculated, these are then used to calculate the context vectors for every instance of the ambiguous word. This is done by calculating the resultant vector of the word vectors of all words in the context of the ambiguous word.

The context vectors are then clustered using a clustering algorithm and a sense vector for each cluster is calculated as the centroid of context vectors of that cluster. We observe that context vectors can be easily used to create a metric that measures "distances" between words based on their meanings from their contexts. We also note that the context vectors represent the meaning of the ambiguous words in their contexts. We would, therefore, expect the context vectors of all the ambiguous words having the same sense to have approximately the same direction in the multidimensional space.

The approximate "distance" between the meanings of two words is measured by finding the cosine of the angle between their vectors, which determines the extent of the overlap of the vectors and measures similarity of concepts. The lengths of the vectors are weighted by log inverse document frequencies of the words. Log inverse document frequency is a concept from the field of Information Retrieval that describes how uniformly a word is distributed over the text documents under consideration. Words such as "idea" or "help" are approximately uniformly distributed throughout all the text documents under consideration and give little information about a specific subject. Words that are localized in a few small areas of the documents usually discuss and relate to certain specific topics. Thus the log inverse document frequency gives an approximate magnitude of the ability of a word to distinguish between different topics. This is similar to the idea of information content described in section 2.

Intuitively, the process described by Schütze puts forth a scale that measures the extent to which two words are related, but not without some pitfalls. Firstly, to improve the accuracy and the reliability of the results we need that vectors be weighted, which requires us to calculate the log inverse document frequencies of all the words in the word-space. Secondly, Schütze observes that the algorithm gives good results when the vectors of the words in the context of the ambiguous word have a high degree of "discriminating potential", i.e. the ability to distinguish between different topics. This implies that the same algorithm may give different results for the same words in different texts.

Some highly related work, using context vectors, has also been done by Inkpen and Hirst [11]. They attempt to disambiguate near-synonyms in text using various "indicators". Near-synonyms are words whose senses are almost indistinguishable. There is only a fine difference between their senses.

Their disambiguation algorithm considered a number of "indicators" to determine the correct sense of the word. The suggestions from the "indicators" were then weighted by a decision tree to get the final result. The decision tree was learnt from a test data set.

One of the "indicators" used in the process was based on context vectors. Using an approach similar to that described by Schütze, context vectors were created for the context of the word and for the glosses of each sense of the target word. The glosses were considered as a bag of words and the word vectors for these words were summed to get the context vectors corresponding to the glosses. The distance between the vector corresponding to the text and that corresponding to the gloss was measured (as the cosine of the angle between the vectors). The nearness of the vectors was used as an indicator to pick the correct sense of the target word.

The use of context vectors described by Schütze gives us a way of describing a concept, using the context it occurs in. It also allows us to imagine a way to augment the short glosses with knowledge from an external source (a corpus of text).

3.2 A Measure of Semantic Relatedness based on Context Vectors

We introduce a measure of semantic relatedness based on context vectors that is inspired by Schütze's approach. In our approach, each concept in WordNet is represented by a *gloss vector*. A gloss vector is essentially a context vector formed by considering a WordNet gloss as the context. The semantic relatedness of two concepts then is simply the cosine of the angle between the corresponding normalized gloss vectors.

In order to create gloss vectors we start by creating a word space, a list of words that would form the dimensions of the vectors. This list of words should contain words that are highly topical, having great potential to discriminate topics. Schütze used frequency cutoffs and the χ^2 test of association on the words of a large corpus. For our experiments we use the WordNet glosses as a corpus, and select content words for the word space from it. We use a list of stop words to eliminate function words. We experiment with different frequency cutoffs and study their effect on the measure.

The next step in creating gloss vectors is the creation of word vectors corresponding to all content words in the WordNet glosses. The process is similar to that described by Schütze. To create a word vector for word w:

1. Initialize the vector to a zero vector $\vec{w}$.

2. Find every occurrence of w in a large corpus.

3. For each occurrence, increment those dimensions of $\vec{w}$ that correspond to the words from the word space, present in a window of context around w in the corpus.

The vector $\vec{w}$, therefore, encodes the co-occurrence information of w. Using this method we create word vectors for all content words present in the WordNet glosses. Again, the content words, whose word vectors are created, can be selected using various means such as frequency cutoffs, stop-lists, etc. For our experiments, we create word vectors from the "WordNet gloss corpus", a corpus composed of all the glosses present in WordNet. We consider each gloss as the context.

Intuitively, we can imagine the multidimensional "word vector space" to be composed of a large number of "pockets of space", each pocket corresponding to a certain topic or aspect of the real world. Depending on their direction, the word vectors are weighted by the various pockets based on the senses of the word (corresponding to the word vector). To illustrate this point, consider the word *bank*, which could mean "a financial institution" or a "river bank". The word vector for *bank* would be weighted by the pockets of the space corresponding to finance and by pockets of the space related to rivers (and nature). This is because the word *bank* would co-occur with words from these two categories, in a large corpus. Other pockets, related to the human body for instance, would have no bearing on this word vector at all. Ideally, every word vector would be weighted by pockets or topics related to the corresponding word (and it senses).

Having created the word vectors, the gloss vector for a WordNet concept is created by adding the word vectors of every content word in its gloss. For example, consider the gloss of *lamp – an artificial source of visible illumination*. The gloss vector for *lamp* would be formed by adding the word vectors of *artificial*, *source*, *visible* and *illumination*.

Again, imagining the "pocket" view of the multidimensional space, for a gloss vector we would hope and expect that the particular pocket in the space would more highly weight that gloss vector as compared to the other pockets. This would be because, the content words in the gloss would have at least one topic in common, and all the word vectors corresponding to these content words would be weighted by one common pocket. The gloss vector would, therefore, be more highly weighted towards this common topic and this common topic describes, to some extent, the concept under consideration.

This formulation of the Vector measure is independent of the dictionary used and is independent of the corpus used, and hence is quite flexible. However, it faces some of the problems faced by Lesk's [15] gloss overlap algorithm for word sense disambiguation – short glosses. To overcome this problem Banerjee and Pedersen [3] adapted their Extended Gloss Overlap measure to use the relations in WordNet to augment the short glosses with other related glosses. We use the relations in WordNet and augment the glosses in a similar way for the vector measure. To create a gloss vector with augmented glosses, consider the gloss of a concept c. With this gloss we concatenate the glosses of all concepts related to c by any WordNet relation. Also, rather than using all the WordNet relations, we can control the speed and efficacy of the measure, by carefully selecting the relations to use for the augmented gloss. The gloss vector for c is then created from the big concatenated gloss. It is also possible to use other dictionaries or representations of the concept to build gloss vectors from.

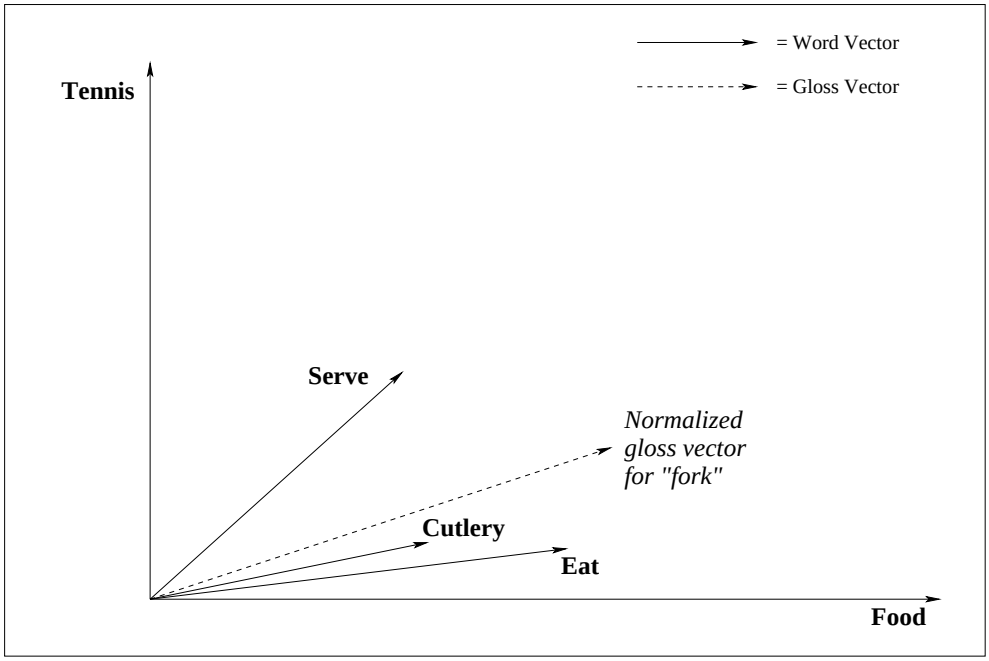

Figure 3: A 2-dimensional vector space showing *word vectors* and a *gloss vector*

The word vectors as well as the gloss vectors usually have a very large number of dimensions (usually tens of thousands) and it is very difficult to visualize this space. Figure 3 attempts to illustrate the vectors in two dimensions (i.e. using a vector space of only 2 dimensions). *Tennis* and *food* are the dimensions of this 2-dimensional space. We see that the word vector for *serve* is approximately halfway between *tennis* and

food, since the word *serve* could mean to "serve the ball" in the context of tennis or could mean "to serve food" in another context. The word vectors for *eat* and *cutlery* are very close to *food*, since they do not have a sense that is related to tennis. The gloss for the word *fork* – "cutlery used to serve and eat food" – contains the words *cutlery*, *serve* and *eat* (and *food*). The gloss vector for *fork* is formed by adding the word vectors of *cutlery*, *serve* and *eat* and *food*. Thus, *fork* has a gloss vector which is heavily weighted towards *food*. *Food* is, therefore, topical of and is related to the concept of *fork*. However, this is a small gloss. Using augmented glosses, we achieve better representations of concepts to build gloss vectors upon.

Gloss vectors for all concepts in WordNet can be computed in this manner. The relatedness of two concepts is then determined as the cosine of the normalized gloss vectors corresponding to the two concepts:

$$related_{vector}(c_1, c_2) = \cos(angle(\vec{v_1}, \vec{v_2})) \tag{21}$$

where c_1 and c_2 are the two given concepts, $\vec{v_1}$ and $\vec{v_2}$ are the gloss vectors corresponding to the concepts and $angle$ returns the angle between vectors. Using vector products we can rewrite the above relatedness formula as:

$$related_{vector}(c_1, c_2) = \frac{\vec{v_1} \cdot \vec{v_2}}{|v_1||v_2|} \tag{22}$$

We now have a measure of semantic relatedness based on WordNet glosses, which is enhanced with information from a large corpus of text. However, it should be pointed out that this measure is not dependent on WordNet glosses, and can be employed with any representation of concepts (such as dictionary definitions), with co-occurrence counts from any corpus.

4 Experimental Procedure

In the earlier sections we described a number of measures of semantic relatedness. We then explained how extended gloss overlaps could be used as a measure of semantic relatedness. This thesis then introduced a new measure based on context vectors in section 3. These measures are compared in this thesis. This section describes the methodology used to compare and contrast the measures. First a comparison of these measures with the human perception of relatedness is performed. The measures are then compared, based on their effectiveness at the task of word sense disambiguation.

4.1 A Human Relatedness Study

Rubenstein and Goodenough [25] performed an experiment whose goal was to explain the basis of the human perception of synonymy. They had human subjects assign degrees of synonymy, on a scale from 0 to 4, to 65 pairs of carefully chosen words. The experiment was repeated by Miller and Charles [19] on a subset of 30 word pairs of the 65 used by Rubenstein and Goodenough. Rubenstein and Goodenough used 15 subjects for scoring the word pairs and the average of these scores was reported. Miller and Charles used 38 subjects in their experiments to score the 30 word pairs. We use the same set of 30 word pairs to perform a comparison of the measures with human perception of semantic relatedness.

The 30 word pairs covered all degrees of semantic relatedness as assigned by humans: 10 highly related pairs (having a score between 3 and 4), 10 pairs having scores between 1 and 3, indicating intermediate relatedness and 10 pairs that were rather unrelated (scored 0 to 1). Table 2 lists the word pairs with the average semantic relatedness scores assigned by human subjects in the Rubenstein and Goodenough experiments, as well as the Miller and Charles experiments. The values have been taken from [19].

We repeat this experiment using our implementation of these measures. We implemented these measures as Perl modules so as to be able to use them in different tasks. We distribute our Perl implementation of these measures on CPAN (Comprehensive Perl Archive Network) under the GPL. They can be freely downloaded from [22]. We used version 0.05 of the `WordNet::Similarity` package for our experiments.

The measures of semantic relatedness give us the relatedness between *word senses*. So as to use them to determine the relatedness of each of the word pairs, we find the relatedness between every combination of

Table 2: Word pairs used in the human relatedness experiment

Word Pairs	R&G		M&C		Word Pairs	R&G		M&C	
	Score	Rank	Score	Rank		Score	Rank	Score	Rank
car - automobile	3.92	1	3.92	2	*crane - implement*	1.66	15	2.37	14
gem - jewel	3.84	2	3.94	1	*journey - car*	1.16	16	1.55	15
journey - voyage	3.84	2	3.58	6	*monk - oracle*	1.10	17	0.91	21
boy - lad	3.76	3	3.82	3	*cemetery - woodland*	0.95	18	1.18	17
coast - shore	3.70	4	3.60	5	*food - rooster*	0.89	19	1.09	18
asylum - madhouse	3.61	5	3.04	9	*coast - hill*	0.87	20	1.26	16
magician - wizard	3.50	6	3.21	7	*forest - graveyard*	0.84	21	1.00	19
midday - noon	3.42	7	3.94	1	*shore - woodland*	0.63	22	0.90	22
furnace - stove	3.11	8	3.11	8	*monk - slave*	0.55	23	0.57	24
food - fruit	3.08	9	2.69	11	*coast - forest*	0.42	24	0.85	23
bird - cock	3.05	10	2.63	12	*lad - wizard*	0.42	24	0.99	20
bird - crane	2.97	11	2.63	12	*chord - smile*	0.13	25	0.02	27
tool - implement	2.95	12	3.66	4	*glass - magician*	0.11	26	0.44	25
brother - monk	2.82	13	2.74	10	*rooster - voyage*	0.08	27	0.04	26
lad - brother	1.68	14	2.41	13	*noon - string*	0.08	27	0.04	26

word senses of the two words in the pair. The maximum relatedness of the word senses is selected as the relatedness of the pair. The word pairs are then arranged in decreasing order of semantic relatedness. The correlation coefficient of this ranking with respect to the Rubenstein and Goodenough ranking is determined. This correlation coefficient is used to compare the measures.

4.2 An Application Oriented Comparison of Relatedness

Lesk [15] described a method for word sense disambiguation based on overlaps of dictionary definitions of word senses. In order to determine the sense of a target word in a given context, this method finds the extent of overlap of the dictionary definition of each sense of the target word with the senses of words in the context of the target word. Overlaps are defined as string matches. For example, a if phrase like "the American President" occurred in two definitions, this would be considered as an overlap. The sense of the target word with maximum overlap with the senses of the words in the context of the target word is selected as the intended sense of the target word. Banerjee and Pedersen [3] adapted this approach to use WordNet as the dictionary and also enhanced the algorithm to extend glosses by using the glosses of concepts related to them in WordNet. The adapted algorithm [2] [3] showed improvement in results over the original algorithm devised by Lesk.

There is one basic hypothesis underlying the original gloss overlap algorithm of Lesk and its adaptation by Banerjee and Pedersen. This hypothesis says that of the senses of the target word, the intended sense is most related to the senses of the words in the context. The extent of overlap of dictionary definitions in both the above approaches, therefore, attempts to measure the relatedness of the senses of the target word with the senses of the context words. In section 2, we introduced this notion of gloss overlaps as a measure of semantic relatedness. One of the contributions of this thesis is the realization that any measure of relatedness can be used in the adapted Lesk algorithm of Banerjee and Pedersen to perform word sense disambiguation.

We use this extended version of the adapted Lesk algorithm to compare the various measures of semantic relatedness. We use each measure of semantic relatedness and perform disambiguation of test data. The results of disambiguation are used for the comparison of the effectiveness of the measures at this task.

A brief description of the word sense disambiguation algorithm follows. Banerjee and Pedersen describe two approaches to determine the correct sense of the target word in the context – the *local* approach and

the *global* approach. These approaches define ways of scoring the senses of the target word, using semantic relatedness.

The following steps describe the local approach to word sense disambiguation:

1. The set T of the *candidate senses* of the target word is first determined. The target word is the word in the context that has to be disambiguated. The candidate senses are the possible senses of the target word, according to WordNet. The part of speech of the target word is usually known and the possible senses are restricted to that part of speech.

2. The set C of senses of the context words is then determined. Set C contains the senses of all words in the *window of context* of the target word, excluding the target word. Window of context is the set of words selected from the context, used for the disambiguation of the target word. For our experiments we select 3 to 5 words from the context, surrounding the target word, as the window of context.

3. The relatedness of each sense of the target word with the senses of the context words is determined. In order to do this, for each element t in set T, the sum of the relatedness of t with each element in set C is assigned as the score for t. Mathematically,

$$score(t) = \sum_{c \epsilon C} sim(t, c) \tag{23}$$

   where $t \epsilon T$ and $sim(c_1, c_2)$ measures the semantic relatedness between the concepts c_1, c_2.

4. The intended sense of the target word is that sense of the target word having the maximum "score" of relatedness with the senses of the context words. It is the element in set T with the maximum score. Mathematically,

$$t_{selected} = \underset{t \epsilon T}{\operatorname{argmax}} \sum_{c \epsilon C} sim(t, c) \tag{24}$$

   where $t_{selected}$ is the selected sense of the target word.

The following steps describe the global approach to word sense disambiguation:

1. The set T of the *candidate senses* of the target word is first determined.

2. For each word w_i in the window of context, excluding the target word, a set C_i containing the possible senses of w_i is created.

3. For each element t from set T:

   (a) An element c_{it} is picked from each of the sets C_i.

   (b) Set X_t is created with these elements, $X_t = \{c_{1t}, c_{2t}, \ldots, c_{nt}, t\}$.

   (c) A *sub-score* of element t is computed as the sum of the relatedness of every possible pair of concepts from X_t.

   (d) The score for element t is then computed as the sum of sub-scores of t from every possible formulation of the set X_t.

The following example illustrates this scoring procedure to compute the score for sense t of the target word.

   (a) Suppose we have 2 words, w_1 and w_2 in the window of context of the target word, each having 2 senses. Then $C_1 = \{c_{11}, c_{12}\}$ and $C_2 = \{c_{21}, c_{22}\}$ are the sets containing the senses of the words w_1 and w_2, respectively.

   (b) Picking one element from each of the sets C_1 and C_2, we create various possible formulations of set X_t. These are enumerated below:

   $X_{t1} = \{c_{11}, c_{21}, t\}$

   $X_{t2} = \{c_{11}, c_{22}, t\}$

   $X_{t3} = \{c_{12}, c_{21}, t\}$

   $X_{t4} = \{c_{12}, c_{22}, t\}$

   (c) For each formulation of set X_t, a sub score is calculated as the sum of the relatedness of all possible pairs of concepts from the set. Therefore, for X_{t1}, we have

$$sub\ score(X_{t1}) = sim(c_{11}, c_{21}) + sim(c_{11}, t) + sim(c_{21}, t) \tag{25}$$

   where $sim(c_1, c_2)$ is the relatedness of concepts c_1 and c_2.

   Similarly, for X_{t2}, X_{t3} and X_{t4}, we have

$$sub\ score(X_{t2}) = sim(c_{11}, c_{22}) + sim(c_{11}, t) + sim(c_{22}, t) \tag{26}$$

$$sub\ score(X_{t3}) = sim(c_{12}, c_{21}) + sim(c_{12}, t) + sim(c_{21}, t) \tag{27}$$

$$sub\ score(X_{t4}) = sim(c_{12}, c_{22}) + sim(c_{12}, t) + sim(c_{22}, t) \tag{28}$$

(d) The score for sense t of the target word is the sum of the sub scores for t. In this case,

$$score(t) = sub\ score(X_{t1}) + sub\ score(X_{t2}) + sub\ score(X_{t3}) + sub\ score(X_{t4}) \quad (29)$$

4. The score for each element in set T is computed. The element from the set T having the maximum score is then the selected sense of the target word.

In both the above approaches, the relatedness of concepts can be found using any of the measures of semantic relatedness. We perform a number of experiments with each of the measures of relatedness, varying parameters of the algorithm such as the local/global approach, window of context, etc. We also vary a number of measure-specific parameters to compare the measures.

5 Description of the Data

To test and evaluate the measures in performing word sense disambiguation, we require a gold standard to measure the performance against. Such data was provided for comparing various Word Sense Disambiguation systems at the SENSEVAL-2 [7] exercise. We use this data to measure the performance of the measures at the task of word sense disambiguation.

The information content based measures are tested using information content values computed from various corpora. A description of these is also provided in this section.

5.1 The SENSEVAL-2 Data

SENSEVAL-2 was an international competition where the entrants evaluated their word sense disambiguation systems on common data in order to compare results in a rigorous way. The data for testing the systems was carefully prepared by lexicographers.

The SENSEVAL-2 competition consisted of two tasks – the *lexical sample* task and the *all words* task. In the *lexical sample* task the participating disambiguation systems were required to disambiguate only a single word in each given context. The *all words* task involved the disambiguation of every content word in the context. Data sets for both these tasks were provided to the teams to measure the accuracy of their systems.

In our experiments we used the lexical sample test data. The lexical sample data consists of 73 different target words. The test data is composed of approximately 4328 instances. There was also training data provided (for supervised systems), but we do not use that data. Each instance is a short paragraph containing 3 to 4 sentences and an occurrence of the target word. The data contains multiple instances for each target word. Also, each target word is used in a single part–of–speech in all instances that it occurs. For example, the target word *chair* occurs as a noun in all instances where *chair* is the target word to be disambiguated.

A typical example of an instance is shown if Figure 4.

The entire instance is enclosed within `<instance>` and `</instance>` tags. Each instance is identified by an ID specified within the tag ("art.30010" in the above example). The target word (*art*) is enclosed within `<head>` and `</head>` tags.

```
<instance id="art.30010" docsrc="wsj_1686.mrg_1">
<context>
After all, farmers here work with "hazardous" chemicals every day, many
of them the same chemicals that would have been destroyed in the
incinerator. We know they are dangerous, but if handled with care, their
benefits far outweigh any risk to the environment. Just because Stamford,
Conn., High School did nothing when its valuable 1930s mural was thrown in
the trash does not mean the city no longer owns the work of
<head> art </head>, a federal judge ruled.
</context>
</instance>
```

Figure 4: Example of an instance from SENSEVAL-2 data

Of the 73 target words, 29 occur as nouns in all their instances, 29 as verbs and 15 as adjectives. 1754 instances contain the noun target words, 1806 contain verb target words and 768 instances contain adjectives as target words. The correct answers to all the instances are provided in a separate key. This key is used for evaluating and scoring our results. This key was not available to the participants until after the event.

In our experiments we compare 7 measures of semantic relatedness. Four of those measures are noun-only measures and cannot measure the relatedness of words in other parts–of–speech. To make the comparison of the measures fair and equivalent, we used only the instances with nouns as target words. However, of the 1754 noun instances, it turned out that in 31 instances the target word did not have noun candidate senses. This was because they were used as *compounds* with surrounding words. Compounds or compound words are multi-word sequences of 2 or more words that behave as a single entity and refer to a concept in WordNet. WordNet synsets contain a large number of compound words. For example, *art gallery* and *private school* are compounds that can be found in WordNet. In 31 instances, the target word formed compounds with surrounding words, and these compounds had no noun senses. These instances were removed from the evaluation set and the set of 1723 instances was used for all the experiments involving the noun-only measures. Table 3 summarizes the characteristics of each target word in this noun data. The table lists all the noun target words. For each word it lists in the first column the number of noun senses in WordNet for that word. The base form of the word is used to get this count. What this means is that even though the in some instances *art* may be used as *arts* we count the number of senses of *art* in WordNet. The second

Table 3: Summary of the lexical sample data set for noun target words

Word	Instances	WordNet Senses	Candidate Senses	Word	Instances	WordNet Senses	Candidate Senses
art	98	4	14	grip	42	7	8
authority	92	7	9	hearth	32	3	3
bar	151	13	21	holiday	31	2	3
bum	44	4	4	lady	53	3	8
chair	69	4	7	material	69	5	10
channel	73	7	13	mouth	57	8	8
child	63	4	5	nation	37	4	6
church	64	3	9	nature	44	5	6
circuit	85	6	15	post	78	8	12
day	134	10	18	restraint	45	6	7
detention	32	2	5	sense	50	5	11
dyke	28	2	2	spade	33	3	4
facility	58	5	7	stress	39	5	5
fatigue	43	4	6	yew	28	2	3
feeling	51	6	7				

column displays the total number of candidate senses that are considered for all the instances of the target word. This value differs from the number of WordNet senses of the word, because the target word may appear in different morphological forms – for example, instances could contain *art* or *arts* as the target word. Compound words containing the target word could also be the target sense in some instances. For example, *art gallery* is a valid target sense in one of the instances. All these different forms of the target word have different senses, and it is up to the word sense disambiguation algorithm to determine the set of candidate senses for the target word.

As separate comparison was done of the measures that could handle all parts of speech. For these measures, all of the 4328 instances (nouns, verbs and adjectives) were used. Table 4 and table 5 list out the verb and

Table 4: Summary of the lexical sample data set for verb target words

Word	Instances	WordNet Senses	Candidate Senses	Word	Instances	WordNet Senses	Candidate Senses
begin	280	10	7	match	42	9	7
call	66	28	17	play	66	35	20
carry	66	39	20	pull	60	18	25
collaborate	30	2	2	replace	45	4	4
develop	69	21	14	see	69	24	13
draw	41	35	22	serve	51	15	11
dress	59	15	12	strike	54	20	20
drift	32	10	9	train	63	11	8
drive	42	21	13	treat	44	8	5
face	93	14	6	turn	67	26	26
ferret	1	3	1	use	76	6	6
find	68	16	17	wander	50	5	5
keep	67	22	20	wash	12	12	7
leave	66	14	10	work	60	27	18
live	67	7	9				

the adjective target words.

Before using the data in experiments, it was preprocessed to make it easier to run experiments on. The following preprocessing steps were taken:

1. Separate files were created for each target word, containing the instances for that target word.
2. XML tags between [and] were removed.
3. Special character codes were removed.
4. Punctuation was removed.
5. The data was converted to lower case.

Table 5: Summary of the lexical sample data set for adjective target words

Word	Instances	WordNet Senses	Candidate Senses	Word	Instances	WordNet Senses	Candidate Senses
blind	55	3	6	green	94	7	14
colorless	35	2	3	local	38	3	4
cool	52	6	7	natural	103	10	23
faithful	23	3	3	oblique	29	2	3
fine	70	9	14	simple	66	7	5
fit	29	3	3	solemn	25	2	2
free	82	8	13	vital	38	4	4
graceful	29	2	2				

6. All compound words were identified in the data.

For evaluating the results of the disambiguation algorithm, the precision is computed for 1723 noun instances when all the measures are being compared and for all the 4238 instances when only the measures that can handle all parts of speech are being compared. The scoring program has been provided by the organizers of SENSEVAL-2. This scoring program considers all the 4328 instances while computing the recall. Since, we are scoring only 1723 instances out of these, the recall value generated by the scoring program is ignored. In the case when the measures attempt to assign a sense tag to all the instances, the precision would be equal to recall and we will call this the accuracy. In the experiments, we also have an option to not attempt an instance if the relatedness score obtained is zero. For these experiments we manually compute the precision and recall.

5.2 Corpora for Computing Information Content

Three of the measures of semantic relatedness use information from large corpora along with WordNet to compute the semantic relatedness of word senses. The corpora are used to compute information content values, which specify the specificity or generality of each concept.

Table 6: Summary of corpora used to compute Information Content

Corpus	Number of Tokens	Number of Types
SemCor 1.7	198,796	23,301
Brown	1,035,651	42,419
Treebank	1,290,000	50,000
BNC	100,106,008	939,0028

In our experiments we used the following corpora for computing information content: SemCor 1.7 [18], the Brown Corpus [9], the Penn Treebank-2 [17] and the British National Corpus. SemCor is a semantically tagged subset of the Brown Corpus. This corpus was used considering the sense tags in one set of experiments and ignoring the sense tags for another set of experiments. From the Treebank corpus, only the plain text "Wall Street Journal" articles were used to compute information content.

6 Results and Analysis

6.1 Human Perception of Relatedness

In order to compare measures of semantic relatedness to human perception of relatedness, we use a set of 30 word pairs. Miller and Charles [19], in an experiment, had human subjects assign scores of relatedness to these word pairs. The word pairs are ranked, based on the assigned scores. The 30 word pairs selected by Miller and Charles was a subset of a set of 65 pairs used by Rubenstein and Goodenough [25], in a similar experiment conducted over 25 years before the Miller and Charles experiment. Scores assigned to the word pairs by the measures of semantic relatedness are used to arrange the word pairs in a ranked list.

Spearman's correlation coefficient is used to determine how close two rankings of a set of data are.The value of Spearman's correlation coefficient ranges from -1 to 1. A value of 1 indicates identical rankings; a value of -1 indicates exactly opposite rankings; a value of 0 indicates no correlation between the rankings. Other values of the coefficient indicate intermediate levels of correlation between these.

It is interesting to note that the correlation between the Miller and Charles ranking and the Rubenstein and Goodenough ranking of the 30 word pairs is approximately 0.95. This shows that there has been little change in the human perception of the semantic relatedness of the 30 word pairs over 25 years and that the experiment is repeatable.

Table 7 summarizes the correlation between the measures of semantic relatedness and the human rankings from the Miller-Charles experiment and from the Rubenstein-Goodenough experiment.

The extended gloss overlaps measure was run with all the WordNet relations and a standard stop-list. The information content based measures, i.e. Resnik, Lin and Jiang-Conrath, were run with the information content from various sources and the best values are shown in the table. The Vector measure was run using all relations, and word vectors computed from the WordNet glosses alone. Various frequency cutoffs were tried to select the dimensions of the vectors. The best results are shown in the table.

We find that the Vector measure corresponds very closely to the human perception of relatedness. This is possibly because the measure attempts to imitate the way humans perceive relatedness. Miller and Charles show that humans build contextual representations of words from their usage in everyday life. The overlap of the contextual representations determines the semantic relatedness of words in the minds of human beings.

Table 7: Correlation between the measures of relatedness and human perception of relatedness

Relatedness Measures	M & C	R & G
Vector	0.877	0.849
Jiang & Conrath (using Treebank)	0.826	0.873
Extended Gloss Overlaps	0.807	0.810
Hirst & St.Onge	0.779	0.810
Resnik (using BNC)	0.771	0.781
Lin (using BNC)	0.760	0.801
Leacock & Chodorow	0.721	0.749

The Vector measure closely follows this pattern. The Jiang-Conrath measure also does very well as a measure of semantic relatedness. This shows that using corpus statistics in the form of information content works well in a relatedness measure. The fact that the extended gloss overlaps also does almost as well indicates that overlaps in the dictionary definitions of words is also a good indicator of the human judgment of relatedness.

We made a number of variations in the Vector measure to see how it performed in different settings. The Vector measure creates gloss vectors from word vectors. We created these word vectors from WordNet glosses. We had approximately 54,000 word vectors, with each vector having approximately 54,000 dimensions. We experimented with frequency cutoffs, to reduce the size of the vectors and the number of word vectors. We used the 20,000 most frequently occurring words for the dimensions of the vectors and we created word vectors for the 2,000 most frequently occurring words. We also tried upper and lower frequency cutoffs. We used words with frequencies of occurrence between 5 and 1,000 as dimensions.

The WordNet relations used for calculating the relatedness were also varied. In the first set of experiments, the glosses from all concepts related to the target concept by any WordNet relation were concatenated with the gloss of the target concept while forming the gloss vector. In the second set of experiments only the glosses of the target concepts were used to form the gloss vectors. Table 8 summarizes the correlation with human perception of the variations in the Vector measure.

Notice that having upper and lower frequency cutoffs has a great effect on the correlation coefficient. How-

Table 8: Variations in the Correlation Coefficients for the Vector measure

Word Vector Dimensions	Relations Used	
	All	Gloss
Words with frequencies 5 to 1,000	0.877	0.620
20,000 most frequent words	0.716	0.517
No frequency cut-offs	0.713	0.571

ever, when no cutoffs are used or only an upper cutoff is used, the correlation is much lower and almost the same for both cases. This suggests that words with very low frequencies add a great deal of noise to the vectors. Also, it suggests that a careful selection of the words for the dimensions of the vectors could improve the measure even more.

Another point to observe is that, using all the WordNet relations, no change in the correlation coefficient is seen between having no frequency cutoffs and having only an upper cutoff. But a lot of variations in the correlation coefficient are seen when only the glosses of the concepts are used. This shows that the size of the glosses is too small to completely describe concepts. Concatenating the glosses of related senses in WordNet provides a more complete description of the target concepts. Building gloss vectors upon less complete description, thus gives us lower correlation values.

As a point of comparison we ran the Extended Gloss Overlaps measure with only the gloss-gloss relation. In this scenario, the Extended Gloss Overlaps measure computes the relatedness of two concepts as the extent of overlap of their glosses. It does not use the glosses of any of the related concepts. With this setting the Extend Gloss Overlaps measure achieved a correlation coefficient of 0.527 with respect to the Miller and Charles ranking. Given the same information, the Vector measure did only slightly better in one case (when word vectors were from WordNet and no frequency cutoffs).

We ran a number of additional experiments to observe variations in the behavior of the information content based measures. Table 9 summarizes the correlation of the information content based measures with different sources of information content. Information content values were computed using frequency counts from SemCor, the Brown Corpus, the Treebank Corpus (WSJ articles) and the British National Corpus. Counting was done using our method of counting and no smoothing of frequency counts was done. Correlation

Table 9: Variations in the Correlation Coefficients for the measures based on Information Content

Information Content Source	Resnik	Lin	Jiang Conrath
SemCor	0.714	0.698	0.727
Brown	0.730	0.744	0.802
Treebank	0.746	0.749	0.826
BNC	0.753	0.751	0.812

coefficients are shown with respect to the Miller and Charles rankings of the word pairs.

SemCor is a sense-tagged subset of the Brown corpus. The words in the corpus have been manually tagged with their appropriate senses by human experts. However, the size of this corpus is relatively small – approximately 200,000 words. The Brown corpus, the Treebank and the BNC on the other hand are plain text corpora, with no annotations. These corpora contain a lot more text than SemCor. The Brown corpus is a corpus of 1,035,651 words; The Treebank is a corpus of 1,290,000 words; The BNC is a corpus of 100,106,008 words.

We observe from the table that all the measures get closer to human perception of relatedness with the increase in the quantity of corpus data used for the calculating information content. It is important to note that sense-tagged text is extremely expensive to create and is much harder to come by. The Jiang-Conrath measure shows the maximum improvement in the correlation coefficient with increase in corpus data, and appears to have approximately the same correlation coefficient with information content computed from different sized corpora. This suggests that the Jiang-Conrath measure is relatively independent of the size of the corpus used for computing information content, as long as the corpus is above a certain minimum size and is a relatively balanced corpus.

We also conducted experiments to observe the effect of Resnik's method of counting and the smoothing of frequency counts, while computing information content. We computed information content from the BNC, with and without Add-1 smoothing. We also computed information content from the BNC, using our method of frequency counting and using Resnik's method of frequency counting. Table 10 summarizes the outcome of these experiments.

Table 10: Effect of smoothing and counting schemes on Correlation Coefficients for the measures based on Information Content

Infocontent from BNC	Resnik	Lin	Jiang Conrath
Our counting, No smoothing	0.753	0.751	0.812
Our counting, Add-1 smoothing	0.752	0.751	0.812
Resnik counting, No smoothing	0.771	0.745	0.790
Resnik counting, Add-1 smoothing	0.771	0.760	0.790

We observe that the Add-1 smoothing does not have an appreciable impact on the correlation coefficients. The Resnik method of counting, however, slightly improves the performance of the Resnik measure and slightly degrades the performance of the Jiang-Conrath measure.

6.2 Application-Oriented Comparison of the Measures

A comparison of the measures with respect to their performance in an application tells us a different story about the measures. The results of the comparison may or may not correspond with those of the human relatedness study, but would give us an idea if using one measure really has any impact on a Natural Language Processing task over another measure.

We compare the measures of semantic relatedness in a Word Sense Disambiguation task. We modify the Adapted Lesk algorithm of Banerjee and Pedersen [3], such that the various measures of semantic relatedness may be used in the scoring process. The basic hypothesis underlying this modification is that the extent of the overlaps of the glosses in this algorithm is an indicator of the semantic relatedness of the two concepts. Based on this hypothesis, in this thesis, we treat extended gloss overlaps as a measure of semantic relatedness. We now attempt to perform Word Sense Disambiguation using the same basic algorithm, only replacing the extended gloss overlaps measure with other measures of semantic relatedness.

The first experiment did a basic comparison of all the measures. A majority of the measures – Resnik, Jiang-Conrath, Lin, Leacock-Chodorow – can only process nouns. In order that we have a fair comparison

Table 11: Comparison of all the measures at WSD on the SENSEVAL-2 noun data

	Window Size = 3		Window Size = 5	
	Local	Global	Local	Global
Jiang & Conrath (using SemCor)	0.447	0.447	0.457	0.453
Extended Gloss Overlaps	0.401	0.399	0.428	0.427
Lin (using SemCor)	0.362	0.351	0.390	0.383
Vector	0.340	-	0.340	-
Hirst & St.Onge	0.304	0.302	0.328	0.319
Resnik (using SemCor)	0.280	0.283	0.287	0.302
Leacock & Chodorow	0.288	0.297	0.282	0.298

of the various measures, we made some modifications to the Word Sense Disambiguation algorithm. We changed the algorithm so that only the noun senses of all words in the context would be considered. Also, only the 29 noun test sets from the SENSEVAL-2 test data were used for the experiment. The words to be disambiguated in the SENSEVAL-2 test set can be found in table 3 in section 5. Table 11 summarizes the disambiguation accuracies that were obtained using each of the measures in various configurations of the disambiguation algorithm. The different configurations were determined by the size of the window of context and the disambiguation strategy (local or global) that was selected.

This Word Sense Disambiguation algorithm was originally created with the extended gloss overlaps measure at the heart of the algorithm. But by looking at table 11 we observe that by replacing the extended gloss overlaps with the Jiang-Conrath measure we get better disambiguation accuracy. Barring the Lin measure and the Extended Gloss Overlaps measure, none of the other measures show much change in disambiguation accuracy across the various configurations of the algorithm. The Vector measure, due to its computational complexity, did not run to completion for the global approach. The Vector measure processes vectors having approximately 50,000 dimensions. Because of the large size of the vectors, all the gloss vectors cannot be loaded into memory. Reading the vectors off the disk really slows down the measure. Due to this time complexity, the experiment didn't run to completion for the global approach. For the local approach it achieved reasonable accuracies.

Table 12: Comparison of three measures at WSD on all of the SENSEVAL-2 data

Measure	Score
Extended Gloss Overlaps	0.340
Vector	0.293
Hirst & St.Onge	0.229

Three of the measures – Hirst-St.Onge, Extended Gloss Overlaps and the Vector measure can handle all parts of speech. The other measures are limited to nouns only. We did a separate comparison of these measure in the word sense disambiguation of all the SENSEVAL-2 test data (all parts of speech). We ran the experiments using the local disambiguation strategy and a window size of 5. Table 12 summarizes these results.

The results indicate that the Extended Gloss Overlaps performs the best at the task on all parts of speech. The Hirst-St.Onge measure seems to do rather poorly. This could be attributed to the fact that the measure uses all the relations in WordNet to compute semantic relatedness. WordNet has a lot richer network of relationships for nouns, as compared to the other parts of speech. So, when dealing only with nouns, the Hirst-St.Onge measure benefits from the denser network for nouns. With all parts of speech, however, it is put at a disadvantage by the sparser network of relations. The Vector measure does not get as good an accuracy as extended gloss overlaps. This is possibly because of the noise that gets incorporated into the vectors from the corpus.

On the whole, the Vector measure corresponds well with the human perception of semantic relatedness. It performs reasonably well at the word sense disambiguation task, though it does not perform as well as we would have hoped. However, the biggest advantage of the measure is that it can handle any part of speech. It is not tied down to WordNet. The measure can be made WordNet-independent by using representations of concepts other than glosses and computing word vectors from other corpora. Also, there is much room for improvement in the performance of the measure, by eliminating noise from the word vectors. This could be done by better selection of the content words forming the dimensions of the vectors (for example, using frequency cutoffs) and by minimizing noise while counting co-occurrences (for example, getting counts for only the most associated words determined using statistical tests of association).

7 Related Work

This section takes a look at some of the related work. We considered some WordNet–based measures of semantic relatedness in this thesis. Some other measures that have been used in applications are also described here. Finally, we look at some supervised and unsupervised approaches to Word Sense Disambiguation.

7.1 Semantic Relatedness

Niwa and Nitta [20] compare the approaches to measuring semantic distance of words – using a large corpus of text, such as the one described by Schütze [26], and using a dictionary. Semantic measures that use a large corpus of text to measure distances between words are hindered because of the insufficient data for rare senses of words. Niwa and Nitta propose a dictionary based approach, that uses a word network such as that in an ordinary dictionary, where each word is linked to other words that occur in its definition (description of its meaning). Therefore, each word is a node in this network. A set of selected nodes, usually corresponding to words having medium range frequencies (51 to 1050) in the Collins English Dictionary, are selected as origins. Vectors corresponding to each word are calculated as the shortest distances of the word to each of the origins. The distance of a word from an origin is measured by counting the number of links between nodes required to be traversed to reach the origin. In a simplistic case the links are assumed to have a fixed weight. But this treats low-frequency words equivalent to high frequency words. However, if we consider the relatively low frequency word *limb* that occurs in the definition of the relatively high frequency word *hand*, we see that these two words (*hand* and *limb*) are more strongly related than *hand* and *hold* (which may also occur in the definition of hand). Elaborating on this, if *hand, limb* and *hold* occur in a sentence we would instantly associate *hand* with *limb* but we cannot say if there is or is not any relation between *hand* and *hold*. Thus, we need to give more weight to links that pass through low frequency words. Taking word frequency into consideration, the following definition of link weight is used:

$$length(W_1, W_2) \underset{def}{=} -log\left(\frac{n^2}{N_1 \cdot N_2}\right) \tag{30}$$

where W_1 and W_2 are the words, N_1 and N_2 are the number of links from the corresponding words and *n* is the number of links between the words.

Niwa and Nitta compare dictionary based vectors with co-occurrence based vectors, where the vector of

a word is the probability that an origin word occurs in the context (in a given window of words) of the word. These two representations are evaluated by applying them to real world applications and quantifying the results. Both measures are first applied to *Word Sense Disambiguation* and then to the *Learning of positives or negatives*, where it is required to determine whether a word has a positive or negative meaning. It was observed that the co-occurrence based idea works better for the *Word Sense Disambiguation* and the dictionary based approach gives better results for the *Learning of positive or negative*. From this, the conclusion is that the dictionary based vectors contain some different semantic information about the words and warrants further investigation. It is also observed that for the dictionary based vectors, the network of words is almost independent of the dictionary that is used, i.e. any dictionary gives us almost the same network. Consequently, we are presented with yet another novel way to think about the semantic relatedness of words.

Budanitsky provides a comprehensive discussion and comparison of a number of methods applied to find semantic distances between words. The discussion includes algorithms using WordNet, as well as those not using WordNet. Working on a network of biomedical literature very similar to that of WordNet, Rada et. al. [23] define a simple edge counting technique that describes the conceptual distance between any two words of the network, and it works well primarily due to the fact that the network of words is made using solely a "Broader-than" (opposite of *is-a* and *part-of*) relationship. In this technique, the number of links between the two words under consideration gives us the degree of relatedness. This simple edge-counting technique is enhanced by Sussna [27] in order for it to apply to all the types of relationships defined in WordNet and for it to take into consideration the reduced edge lengths at higher depths in the WordNet network hierarchy. For example, *entity* is the most abstract class at the top of the *is-a* hierarchy and the degree of specificity increases as we go lower down in the hierarchy. We see that as we go lower down in the hierarchy the nodes are more related to their parent nodes than the nodes higher up in the tree. *Nickel* is more related to *coin* (both lower down in the hierarchy) than the degree to which *living being* and *phytoplankton* are related. Sussna achieves this by assigning specific weights to different types of relationships in WordNet and by applying a "depth scaling factor" that scales the weight corresponding to the depth of the edge within the WordNet network hierarchy. Apart from these Budanitsky also describes and evaluates other approaches to semantic relatedness using other data sources, such as the *Longman Dictionary of Contemporary English* (LDOCE) and *Roget's Thesaurus* available in electronic format.

Budanitsky compares and evaluates all these various measures with respect to certain human judged data and applying these to solve a problem of detecting and correcting malapropisms in selected text and assessing the results. Malapropisms are spelling errors that result in another word that is not supposed to be in the text at that given point. It was found that the method described by Agirre and Rigau consistently gave better results than the baseline approach of assigning the most common sense of the ambiguous word and was described as "promising". The application of the proposal by Rada et. al. to information retrieval demonstrated improvement in performance.

7.2 WordNet–based Methods of Word Sense Disambiguation

Agirre and Rigau [1] develop a notion of conceptual density which is used as the core idea behind their algorithm for Word Sense Disambiguation. They use the context of a given word along with the hierarchy of *is-a* relations in WordNet to determine the correct sense of the word. It proceeds by dividing the hierarchy network of *WordNet* into subhierarchies and each of the senses of the ambiguous word belongs to one subhierarchy. The conceptual density for each subhierarchy is then calculated using the conceptual density formula which, intuitively, describes the amount of space occupied by the context words in each of the subhierarchies. The formula returns, for each sense, the ratio of the area occupied by the subhierarchies of each of the context words within the subhierarchy of the sense to the total area occupied by the subhierarchy of the sense. The sense with the highest conceptual density value is then selected as the correct sense. For example, if a word W has 3 senses, and there are 5 context words (w_1, w_2, ..., w_5) within the context window, having 2 senses each. We divide the hierarchy into 3 subhierarchies, each subhierarchy containing one sense of W. Now we find the conceptual density of each subhierarchy, by determining which subhierarchy contains a greater number of senses of the context words per unit node. The sense of W in the subhierarchy with the highest conceptual density is the selected sense. Budanitsky [5], in his evaluation of this technique, suggests that this notion of conceptual density could be extended to measure semantic relatedness between words. However, he does not specify an exact method to do so.

7.3 Other Approaches to Word Sense Disambiguation

A number of other approaches to Word Sense Disambiguation, not using WordNet, have also been proposed by researchers.

An approach that uses the *Longman Dictionary of Contemporary English* (LDOCE) is described by Kozima and Furugori [13]. They use an approach called "Spreading Activation". Here an activity measure is calculated, starting from the first word to all connected words in the densest part (LDV) of the dictionary and an activity pattern is created. Finally, using the activity value of the target word (w.r.t. the starting word) and a significance value of the target word, a measure of relatedness can be calculated. This formula is slightly refined to include the "fringe" words of the dictionary.

Another approach to Word Sense Disambiguation is taken by Yarowsky [28]. Two properties of the words associated with the ambiguous word are used. Firstly, words near the ambiguous word consistently indicate the correct sense of the word. Secondly, the sense of the ambiguous word in a particular document or discourse remains the same throughout. In order to start the algorithm, a relatively small number of the instances (typically 2% to 15%) of the ambiguous word are hand tagged with their correct senses along with their collocation. With this training set as a starting point, an iterative process is followed where new collocations are first found in the tagged text and these are then applied to the untagged text which is then tagged with the corresponding sense. During this process the second property, i.e. assigning tags based on the discourse, is optionally used to extend the process and to correct erroneously assigned tags. Yarowsky evaluates this method using data extracted from a 460 million word corpus, covering a wide range of texts including news articles, novels, scientific texts, etc. The algorithm out does supervised algorithms in many cases.

8 Conclusions

The starting point of this thesis is found in the work of Banerjee and Pedersen [3], who adapted Lesk's word sense disambiguation algorithm to the lexical database WordNet. In this thesis, we suggested that the method of disambiguation developed by Banerjee and Pedersen could employ any measure of semantic relatedness, not just their method of using extended gloss overlaps. As a result of extending their algorithm, we also came to view extended gloss overlaps as a generic measure of semantic relatedness. We showed that this view was reasonable by carrying out a study of word sense disambiguation using a wide range of measures of semantic relatedness, and obtaining reasonable results. In particular, we observed that the gloss overlap method of Banerjee and Pedersen, and the information content based measure of Jiang and Conrath, resulted in more accurate disambiguation than the other measures [21].

Based on these results, we developed a new measure of semantic relatedness that represents concepts using context vectors, and is then able to establish relatedness by measuring the angle between these vectors. This measure combines the information from a dictionary with statistical information derived from large corpora of text. It was designed to merge the advantages of gloss overlap and information content measures such as those of Banerjee and Pedersen and of Jiang and Conrath, and to do so in such as way as to avoid any strict dependence on the use of a particular resource such as WordNet. In this thesis we evaluated our new measure relative to six other existing measures in two different settings. First we compared all of these measures with respect to human judgments of relatedness, and then as a part of a word sense disambiguation experiment.

For the human relatedness study, we used published results from Rubenstein and Goodenough [25] and Miller and Charles [19]. These two groups of researchers conducted studies of human judgments of relatedness using 30 pairs of words with a large number of subjects, and arrived at relatively consistent results despite the passage of more than 25 years between the studies.

We found that the context vector measure (Vector) correlates extremely well with the results of these human studies, and this is indeed encouraging. We believe that this is due to the fact that the context vector is making its relatedness judgments based on the actual context in which words occur, while the other measures rely more on the existing structure of relations within WordNet to draw their conclusions. This measure can be tailored to particular domains depending on the corpus used to derive the co-occurrence matrices, and makes no restrictions on the parts of speech of the concept pairs to be compared. This is not true of most

of the other measures. The Resnik, Jiang and Conrath, Lin, Leacock and Chodorow measures are limited to studying noun-noun concept pairs, which we believe is overly restrictive.

We then evaluated this measure in the context of a comparative word sense disambiguation experiment, where it did not fare quite as well as did other measures. In particular, it appears that extended gloss overlaps continue to do very well in that domain. However, this is not terribly surprising in that extended gloss overlaps are based on dictionary content, in fact the content of the same dictionary that is used in the disambiguation exercise. Context vectors are not quite as tailored to the dictionary, which while that appears to be a liability with respect to disambiguation, may mean that it has wider applicability.

An additional aspect of this work was the creation of a package of freely available software that implements all of the measures of relatedness discussed in this thesis. This system is known as WordNet::Similarity, and is available through the Comprehensive Perl Archive Network (CPAN). There are two great benefits to creating such a resource. First, to implement these measures required that they be fully understood, and in fact we realized that there were various limitations to the existing measures that we addressed. We improved the handling of concepts with zero information content in the Jiang-Conrath and Lin measures, proposed an alternate counting scheme for Resnik's information content measure, and adapted the Hirst-St.Onge measure to perform with concepts rather than surface forms of words. Second, we gained the benefit of having the feedback from outside users who have downloaded the code, and in effect acted as critics and testers of the package. As a results of their efforts, we feel particularly confident that our implementation and the ensuing results we report are reasonably sound.

9 Future Work

An analysis of the various experiments conducted gives us a number of ideas about future research that would help improve automatic computation of semantic relatedness. We start by studying the measures in their current form and imagine ways of improving their performance.

Analysis of the Word Sense Disambiguation results gives us some insight into the shortcomings of the disambiguation algorithm. Our algorithm for disambiguation is based on the hypothesis that the correct sense of the target word is highly related to senses of words in the context. Due to the heavy computation involved in computing semantic relatedness between word senses, we consider a window of context of size up to 5 content words immediately surrounding and including the target word. We hope to improve disambiguation by selecting this window of context in a more principled manner. Ideas for getting better context words are discussed here.

Some ways of extending the utility of these measures into other domains, such as that of Medical Informatics, are also discussed in this section.

9.1 Extending Gloss Overlaps

Lesk [15] describes a method for Word Sense Disambiguation based on the dictionary definitions of words. He selects that sense of the target word whose dictionary definition has maximum overlapping phrases with definitions of surrounding words in the context. Banerjee and Pedersen [3] implemented this idea using WordNet as the dictionary. They then adapted this method to incorporate the vast amount of information encoded in the synset relations in WordNet. In this thesis we introduced this Adapted Lesk algorithm as a measure of semantic relatedness.

To measure semantic relatedness, the adapted algorithm considers the overlap of glosses of the all synsets related by one link to the two word senses under consideration. Figure 5 shows a schematic of how the extended gloss overlaps measure uses the relations in WordNet to improve upon the basic Lesk algorithm.

In determining the relatedness of `car#n#1` and `bike#n#1`, the algorithm considers all synsets directly related to the two senses by WordNet relations and looks for the extent of overlaps of the glosses of these. The question to ask here would be – why does the measure consider just direct relations? In figure 5

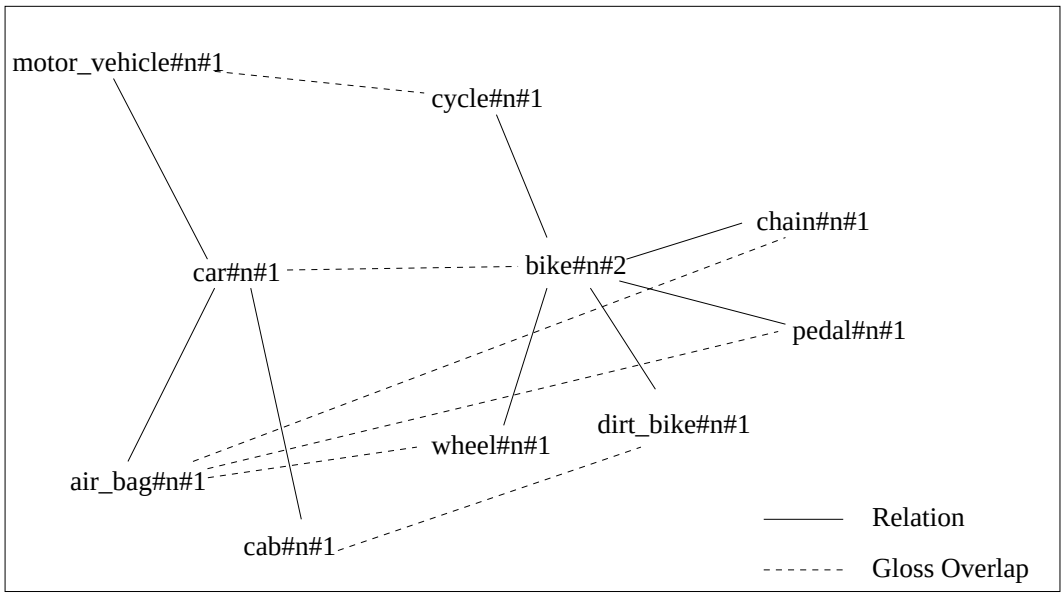

Figure 5: Schematic of the extended gloss overlaps measure

including the gloss of an indirect relation like the *antonym* of the *hypernym* might work equally well or better.

One way to extend the gloss overlaps uses just the *is-a* hierarchy. In this method, while looking for gloss overlaps between two synsets, look for overlaps between all subsuming synsets in the *is-a* hierarchy as well. So here we look at synsets that are more than one link away. We could weight the overlaps of these by the inverse of the distance from the target synsets.

9.2 Refining Gloss Overlaps

Another problem with the Adapted Lesk measure is that the scores generated due to the overlaps mostly comprise of overlaps of non-content words like *and*, *is*, etc, or the overlaps contain only very general and common words like *usually* or *describe*. These contribute very little to the score and just add noise to the measure. An overlap of a highly topical content word is weighted equally with these common word overlaps.

What we would want to do is to weight overlaps of highly specific or topical words higher than overlaps of general or non-topical words. This follows intuitively from the fact that two synsets which are related are very likely to have highly topical words common in their dictionary definitions. For example, concepts like

scuba-diving and *water-sport*, should have terms like *water* common to their definitions.

This does not mean that definitions that do not have highly topical words in their overlaps cannot be highly related. The highly topical words just give a strong evidence of the fact. In order to weight topical words higher than other words during the dictionary definition overlaps, again we could use information content of the senses of the words to weight the overlaps. Words with higher information content are more specific and topical and would weight the score higher than the more general terms in the overlaps.

9.3 Alternate Approaches to creating Word Vectors

Our analysis of the results from experiments on the Vector measure shows that the measure is highly sensitive to the word vectors computed. We first created word vectors without any restrictions and the analyzed the performance of the Vector measure. We then used various frequency cutoffs on the dimensions of the word vectors. The correlation of the Vector measure with human perception greatly improved with carefully selected frequency cutoffs. This shows that if we were could select the dimensions of the vectors in a principled manner, we could hope for further improvement in the performance of the Vector measure and, at the same time, would reduce the size of the vectors. Reduction in the size of the vectors would definitely speed up the performance of the Vector measure.

Some ideas for selection of the dimensions of the word vectors follow:

1. The first method as used and compared in this thesis is that of frequency cutoffs. Very high frequency words tend to be very general words, not specific to any particular topic. The Vector measure works best if the words that form the dimensions of the vectors are highly topical content words. At the same time very low frequency words are very rare and thus contribute to a very small extent to the contextual description of the word or concept. Also, these words form the bulk of the dimensions of the vectors. Thus, eliminating these dimensions should have little effect on the accuracy of the measure, while greatly improving the speed of the measure. Thus, choosing frequency cutoffs carefully is very beneficial.

2. Word vectors may also be created by using statistical tests of association on the words of a corpus to determine how much effect each word has on another. Tests such as the *log likelihood* or *mutual information* may be used. The values of association of the words could be substituted for the co-

occurrence frequencies in the vectors. Intuitively these tests tend to indicate the extent of contextual similarity of words, justifying this approach. Frequency cutoffs could be applied to these as well.

3. The co-occurrence frequencies forming the word vectors could be weighted by the a value indicating the specificity of the co-occurring words. A more topical word occurring in the context of word w says more about the context of w than does a less topical word. Thus, frequencies could be weighted by tf/idf values to take topicality into consideration.

All of these or some combination of these could be used to carefully create the word vectors.

9.4 A Principled Approach to Context Selection

In our word sense disambiguation experiments, we select a window of context of up to 5 content words immediately surrounding the target word. We then determine the relatedness of each sense of the target word with senses of the selected words. The sense of the target word that is most related to the senses of these words is selected as the correct sense of the target word. Intuitively, we would expect that the context or the entire discourse is about, or is related to, a particular topic. So we would expect the content words in this discourse to be specific to that particular topic as well. With this conjecture, the correct sense of the target word would also be related to the topic and hence related to the topical content words around it.

For example, suppose we have a discourse that describes the state of the share market. An instance of the word *bank* is then, in all likelihood, talking about the "financial institution" sense of the word. Further, given the topic of the discourse, we would expect the discourse to contain a lot of other monetary and financial terms such as *money*, *finance*, *credit*, etc. The "financial institution" sense of *bank* would then be highly related to these terms and would be correctly disambiguated by our algorithm.

The reality however is that in a large number of cases the words selected in the window of context have very general senses (not pertaining to any topic in particular) and have no senses related to the correct sense of the target word. For example, words like *further*, *contain*, etc would figure in the 5 words surrounding the target word. The correct sense of the target word is, perhaps, related to the sense of a distant word in the context.

We describe two principled approaches to selecting the window of context that may improve results.

9.4.1 Using Information Content

We notice that in a lot of cases the words immediately surrounding the target word are very general words. As such, these words do not have anything to do with the broad topic of the discourse. However, if we step further away on either side of the target word, we would expect to find words that are more topical and have senses that are more related to the topic of discussion of the context.

An extract from one of the instances from the SENSEVAL-2 data is shown below:

> The barman was back (a leap) at his station behind the <head> bar </head>, as if nothing had happened.

The target word to be disambiguated is enclosed within the <head> and </head> tags. In this sentence, *barman* is most related to the correct sense of the target word. It is at a relatively greater distance from the target word as compared to the some of the other content words. According to our algorithm, *barman* would neither get considered in a window of context of size 3 nor a window of context of size 5 around the target word. On the other hand, words like *behind*, *nothing* and *happen* would be used to disambiguate *bar*. In such cases, the algorithm is put at a disadvantage by the selected words.

Another example is shown below:

> In the middle of the room was a round table covered with oilcloth, and four high-backed carved <head> chairs </head> set around it.

Again in this case, the word *table*, that gives away the correct sense of *chair* in this sentence, is a great many words away from the target word and hence would not figure in the disambiguation process. Instead, words like *set* and *four* would be used by the algorithm.

We wish to look for words in the context that have senses which are more specific to a particular topic. The information content of a word sense gives us exactly this information. A method to compute information content of concepts from a large corpus was introduced by Resnik [24] and is described in section 2 of this thesis. Extending the idea of information content to words, we could select those words of the context that have a sense with high information content. A down-side of doing this would be that even very general

words having some obscure senses would get considered due to the high information content of an obscure sense. For example, the word *object* has 4 noun senses, including senses like "a grammatical constituent that is acted upon", which is highly specific as compared to "a tangible and visible entity". One way to guard against this would be to consider only the information content of the first sense of the word during the selection. This is based on the heuristic that the first sense of a word is the most widely used sense and hence would be the most general sense out of all the senses.

The words from the context that would then be used by the algorithm to disambiguate the target word can be selected by considering only those words whose information content is above a certain cutoff. The cutoff can be empirically determined by performing experiments to see the kind of words that get selected using different values for the cutoff.

As an alternative to selecting *words* from the context, can we instead select *word senses* from the context and use those to determine the correct sense of the target word? Currently, the algorithm attempts to disambiguate the target word by first selecting a set of content words from the context, immediately surrounding the target word. This set of words, along with the target word, is defined as the *window of context*. The algorithm then measures the relatedness of each sense of the target word with every sense of every other word in the window of context. Now instead of selecting a set of words from the context, the algorithm could select a set of word senses. It could consider the context to be a set of word senses, consisting of every sense of every content word in the given instance. The window of context could then be defined as the set of word senses from the context, whose information content value is above a certain cutoff and lie closest to the target word. This would eliminate a number of unnecessary word senses that lie near the target word and also be able reach out and touch words that lie relatively far away from the target word.

9.4.2 Using Lexical Chains

Another method that could potentially be used to select a good set of context words is based on the concept of *lexical chains*. One of the measures of semantic relatedness that was used in the disambiguation experiments, viz. the Hirst-St.Onge measure, was actually designed by Hirst and St.Onge [10] to detect lexical chains in text. They say that if text is "cohesive and coherent", then successive words in the text are related to preceding words and these words form "cohesive" chains in text, which they define as *lexical chains*. Their paper lays down the rules that define what constitutes related words. They then describe a method that uses

this definition of relatedness to create the lexical chains in text.

We propose, as future work, using the notion of lexical chains to restrict the context words considered by the algorithm to those that actually matter for disambiguation.

Once we have a method to detect lexical chains in text, the method to restrict the window of context using this method is straightforward. We would start by creating all the possible lexical chains in the given context. Then, we would select only those chains that pass through the target word. Now, starting from the target word follow these chains backwards as well as forwards and select the nearest words on these chains as the window of context. The disambiguation algorithm would then proceed, as usual, by finding the relatedness of the senses of the target word with the senses of these selected words.

Using this method, we hope to rope-in highly related words, relatively distant from the target word, into the window of context. Figure 6 shows an example of what a lexical chain may look like in a given context.

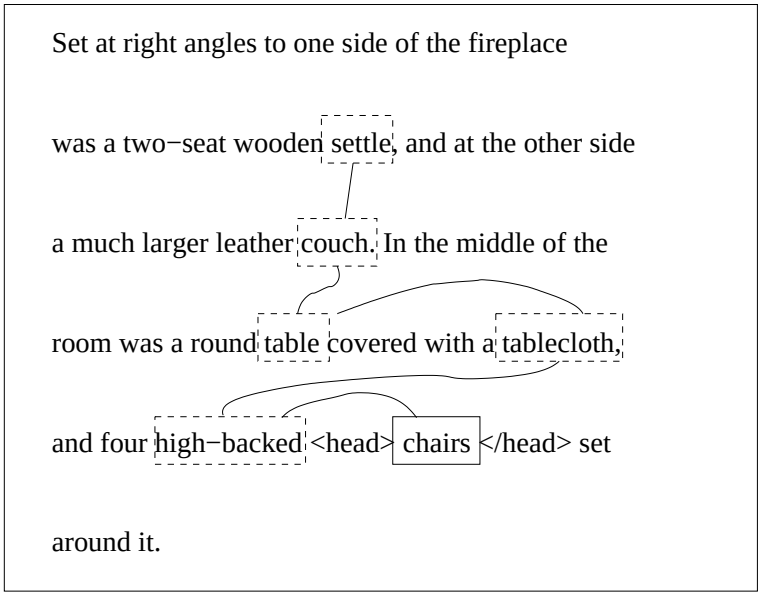

Figure 6: An example of a lexical chain in given context

9.5 Use of Semantic Relatedness in Medical Informatics

A number of taxonomies and semantic networks similar to WordNet exist in other domains. For example, MeSH and SNOMED CT are widely used taxonomies of medical terms used in the Medical Informatics domain. These are comparable in size with WordNet. Other resources such as Clinical Data and Prescriptions are available to hospitals and Clinics. Such resources could be used to measure semantic relatedness of medical terms. If the measures could be modified to take advantage of the various available resources in other domains, these might prove useful for research in these fields.

Such work is being currently carried out by us at the Mayo Clinic in Rochester. We are re-implementing the measures to use a publicly available taxonomy of medical terms called SNOMED. The vast store of Clinical Notes at the Clinic form the source of information content and word vectors for the measures. Being a highly specialized domain, Medical Informatics suffers to lesser extent the problems of ambiguity faced by ordinary English. We hope to capitalize on this advantage of the field. Results from these experiments will be reported soon.

Table 13: An example demonstrating the usage of Spearman's Correlation Coefficient

Elements	Ranking #1	Ranking #2
red	1	2
blue	2	3
green	3	1
yellow	4	4

A Spearman's Rank Correlation Coefficient

Spearman's coefficient is used to determine the similarity between two rankings of the same list of elements. If the two rankings are exactly the same, the Spearman's correlation coefficient between these two rankings is 1. While an exactly reverse ranking gets a value of -1. When the there is no relation between the rankings, the Spearman's correlation coefficient is 0 in such a case. The Spearman's correlation coefficient is computed by the following formula:

$$r = 1 - \frac{6 \cdot \sum_{i=1}^{i=n} D_i^2}{n \cdot (n^2 - 1)} \tag{31}$$

where r is the correlation coefficient, n is the number of elements and D_i is the difference between the ranks for each element in the two rankings.

For example, if two rankings of the elements $\{$*red, blue, green, yellow*$\}$ is as specified in table 13, then using the above formula (equation 31) we can compute the correlation between the rankings as:

$$\begin{aligned} r &= 1 - \frac{6 \cdot (D_{red}^2 + D_{blue}^2 + D_{green}^2 + D_{yellow}^2)}{4 \cdot (4^2 - 1)} \\ &= 1 - \frac{6 \cdot [(1-2)^2 + (2-3)^2 + (3-1)^2 + (4-4)^2]}{60} \\ &= 0.4 \end{aligned}$$

References

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

[2] S. Banerjee. Adapting the Lesk algorithm for word sense disambiguation to WordNet. Master's thesis, Dept. of Computer Science, University of Minnesota, Duluth, 2002.

[3] S. Banerjee and T. Pedersen. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, February 2002.

[4] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, August 2003.

[5] A. Budanitsky. Lexical semantic relatedness and its application in natural language processing. Technical Report CSRG-390, University of Toronto, Department of Computer Science, August 1999.

[6] A. Budanitsky and G. Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, June 2001.

[7] P. Edmonds and S. Cotton, editors. *Proceedings of the Senseval–2 Workshop*. Association for Computational Linguistics, Toulouse, France, 2001.

[8] C. Fellbaum, editor. *WordNet: An electronic lexical database*. MIT Press, 1998.

[9] W. Francis and H. Kucera. *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin, 1982.

[10] G. Hirst and D. St. Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum, editor, *WordNet: An electronic lexical database*, pages 305–332. MIT Press, 1998.

[11] D. Inkpen and G. Hirst. Automatic sense disambiguation of the near-synonyms in a dictionary entry. In *Proceedings of the 4th Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2003)*, pages 258–267, Mexico City, February 2003.

[12] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings on International Conference on Research in Computational Linguistics*, Taiwan, 1997.

[13] H. Kozima and T. Furugori. Similarity between words computed by spreading activation on an english dictionary. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics (EACL-93)*, pages 232–239, Utrecht, 1993.

[14] C. Leacock and M. Chodorow. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An electronic lexical database*, pages 265–283. MIT Press, 1998.

[15] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In *Proceedings of SIGDOC '86*, 1986.

[16] D. Lin. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, Madison, Wisconsin, August 1998.

[17] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[18] G. Miller, C. Leacock, T. Randee, and R. Bunker. A semantic concordance. In *Proceedings of the 3rd DARPA workshop on Human Language Technology*, pages 303–308, Plainsboro, New Jersey, 1993.

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

[20] Y. Niwa and Y. Nitta. Co-occurrence vectors from corpora versus distance vectors from dictionaries. In *Proceedings of the Fifteenth International Conference on Computational Linguistics*, pages 304–309, Kyoto, Japan, 1994.

[21] S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-03)*, Mexico City, Mexico, February 2003.

[22] S. Patwardhan and T. Pedersen. WordNet::Similarity modules version 0.05. Released, 2003. http://search.cpan.org/dist/WordNet-Similarity.

[23] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30, 1989.

[24] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, August 1995.

[25] H. Rubenstein and J.B. Goodenough. Contextual correlates of synonymy. *Computational Linguistics*, 8:627–633, 1965.

[26] H. Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

[27] M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the Second International Conference on Information and Knowledge Management (CIKM-93)*, pages 67–74, Arlington, VA, 1993.

[28] D. Yarowsky. One sense per collocation. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 266–271, 1993.