Maximizing Semantic Relatedness to Perform Word Sense Disambiguation

Ted Pedersen [a,*] Satanjeev Banerjee [b] Siddharth Patwardhan [c]

[a]*Department of Computer Science, University of Minnesota, Duluth, MN*

[b]*Language Technologies Institute, Carnegie-Mellon University, Pittsburgh, PA*

[c]*School of Computing, University of Utah, Salt Lake City, UT*

Abstract

This article presents a method of word sense disambiguation that assigns a target word the sense that is most related to the senses of its neighboring words. We explore the use of measures of similarity and relatedness that are based on finding paths in a concept network, information content derived from a large corpus, and word sense glosses. We observe that measures of relatedness are useful sources of information for disambiguation, and in particular we find that two gloss based measures that we have developed are particularly flexible and effective measures for word sense disambiguation.

Key words: word sense disambiguation, semantic relatedness, semantic similarity

1 Introduction

Word sense disambiguation is the process of assigning a meaning to a particular word based on the context in which it occurs. Often the set of possible meanings for a word is known ahead of time, and is determined by the sense inventory of a Machine Readable Dictionary or lexical database.

When word sense disambiguation is cast as the problem of selecting a sense from an existing inventory, there are at least two different methodologies that can be applied. One option is *supervised learning*, where a system is trained with manually created examples of correctly disambiguated words in context.

---

* Corresponding Author.
  Email address: `tpederse@umn.edu` (Ted Pedersen).
  URL: `http://www.d.umn.edu/~tpederse` (Ted Pedersen).

While supervised approaches have been very popular in recent years, there is no clear means of creating the large amounts of sense tagged text they require to be deployed on a wide scale. Thus, we believe that dictionary based approaches merit continued attention. These methods treat a dictionary or similar resource as both the source of the sense inventory as well as a repository of information about words that can be exploited to distinguish their meanings in text. This work utilizes the lexical database WordNet, but both the disambiguation algorithm we introduce and the the measures of relatedness we describe are well suited for other such resources.

This article presents a method of word sense disambiguation that assigns a sense to a target word by maximizing the relatedness between the target and its neighbors. We carry out disambiguation relative to the senses defined in the lexical database WordNet, and we use both its networked structure and glosses of word meanings to measure semantic relatedness of word senses. Our method is not supervised, and does not require any manually created sense–tagged training examples.

Banerjee and Pedersen [1] began this line of research by adapting the Lesk algorithm [2] for word sense disambiguation to WordNet. Lesk's algorithm disambiguates a target word by selecting the sense whose dictionary gloss shares the largest number of words with the glosses of neighboring words. As this work progressed, we noted (as did Resnik [3]), that gloss overlaps can be viewed as a measure of semantic relatedness. Patwardhan, Banerjee and Pedersen [4] observed that disambiguation can be carried out using any measure that is able to score the relatedness between two word senses. This article represents a generalization and improvement upon that earlier work.

The underlying presumption of our method of disambiguation is that words that occur together in a sentence should be related to some degree. This is not a new observation, nor is it likely to stir up much controversy. What remains unclear is how to best measure semantic relatedness, and which measures will prove most effective in carrying out word sense disambiguation.

In this article present our algorithm, and evaluate it using nine different measures of semantic relatedness, including those of Lesk [2], Wu and Palmer [5], Leacock and Chodorow [6], Hirst and St. Onge [7], Resnik [3], Jiang and Conrath [8], Lin [9], Banerjee and Pedersen [10], and Patwardhan and Pedersen [11].

This article is organized as follows. There is a rich history of dictionary based approaches, and we review several representative approaches that measure semantic relatedness or carry out disambiguation using dictionary content. Then we introduce our algorithm that performs disambiguation by maximizing semantic relatedness. We provide a brief introduction to WordNet, the source of

2

our sense inventory and the knowledge source for the measures of semantic relatedness employed. Then we will describe all nine of the measures of semantic relatedness that we have applied to word sense disambiguation. We present an extensive experimental evaluation using the Senseval-2 English lexical sample data. We believe that our algorithm and its evaluation is noteworthy in that it separates the relatedness measure from the disambiguation algorithm, meaning that any measure of relatedness can be applied. Ultimately this evaluation shows that the extended gloss overlap measure of Banerjee and Pedersen fares well across all parts of speech, although we also observe excellent performance on nouns and verbs by the information content based measure of Jiang and Conrath. Finally, we will conclude with some discussion of these results, and suggestions for future work.

2   Previous Work in Relatedness and Disambiguation

The underlying idea in this article is that semantic relatedness can be used to determine the meaning of words in text. There is a rich history of research in two distinct areas that we draw upon. First, there has been work that exploits glosses of word meanings as found in Machine Readable Dictionaries. Second, networked or hierarchical arrangements of concept information have been utilized as sources of information for word sense disambiguation. We have attempted to merge these two schools of thought into our new gloss based measures, extended gloss overlaps and gloss vectors, that will be described shortly.

2.1   Machine Readable Dictionaries

Dictionaries have long been recognized as possible sources of information for computational methods concerned with word meanings. For example, in the early to mid 1960's, Sparck–Jones [12] developed techniques that identified synonyms by clustering terms based on the content words that occurred in their glosses.

In the mid to late 1960's, Quillian [13] described how to use the content of a machine readable dictionary to make inferences about word meanings. He proposed that the contents of a dictionary be represented in a semantic network. Each meaning associated with a word is represented by a node, and that node is connected to those words that are used to define the concept in the dictionary. The content words in the definitions are in turn connected to the words that are used to define them, and so forth, thus creating a large web of words. Once this structure is created for a variety of concepts, spreading

activation is used to find the intersecting words or concepts in the definitions of a pair of words, thus suggesting how they are related. For example, in one of Quillian's examples he finds that *cry* and *comfort* share the word *sad* in their glosses, which suggests that they are related to this emotion. As such this represents an early use of exploiting gloss overlaps (shared words in dictionary definitions) to make determinations about word meanings.

Due to the limitations of available computing hardware, and the lack of online dictionaries, progress in exploiting dictionary content automatically was slow but steady. However, by the 1980's computing resources were much more powerful, and Machine Readable Dictionaries were becoming more widely available. The Lesk algorithm [2] may be identified as a starting point for a resurgence of activity in this area that continues to this day.

The Lesk algorithm selects a meaning for a particular target word by comparing the dictionary definitions of its possible senses with those of the other content words in the surrounding window of context. It is based on the intuition that word senses that are related to each other are often defined in a dictionary using many of the same words.

In particular, the Lesk algorithm treats glosses as unordered bags of words, and simply counts the number of words that overlap between each sense of the target word and the senses of the other words in the sentence. The algorithm selects the sense of the target word that has the most overlaps with the senses of the surrounding words.

For example, suppose we wish to disambiguate *bank*, in the sentence *I sat on the bank of the lake*. Suppose that $bank_1$ is defined as *financial institution that accepts deposits and channels the money into lending activities*, and $bank_2$ is defined as *sloping land especially beside a body of water*. Suppose that *lake* is only defined with one sense, *a body of water surrounded by land*. There are no overlaps between $bank_1$ and the sense of *lake*, but there are are two content words that overlap between *lake* and $bank_2$, *body* and *water*. Thus, the Lesk algorithm would determine that $bank_2$ is the appropriate sense in this context. One of the innovations of our extended gloss overlap measure is that it takes into account phrasal matches and weights them more heavily than single word matches.

Lesk's description of his algorithm includes various ideas for future research, and in fact several of the issues he raised continue to be topics of research even today. For example, should the Lesk algorithm be used to disambiguate all the words in a sentence at once, or should it proceed sequentially, from one word to the next? If it did proceed sequentially, should the previously assigned senses influence the outcome of the algorithm for following words? Should words that are located further from the target word be given less importance than those

that are nearby? Finally, Lesk also hypothesized that the length of the glosses is likely to be the most important issue in determining the success or failure of this method.

Following on this last point, Wilks et. al. [14] were concerned that dictionary glosses are too short to result in reliable disambiguation. They developed a context vector approach that expands the glosses with related words, which allows for matching to be based on more words and presumably result in finer grained distinctions in meaning than is possible with short glosses. As become standard for much of the work in the early 1990's, they used Longman's Dictionary of Contemporary English (LDOCE). One of the appeals of LDOCE for gloss matching work is that it has a controlled definition vocabulary of approximately 2,200 words, which increases the likelihood of finding overlaps among word senses.

They treat the LDOCE glosses as a corpus, and build a co–occurrence matrix for the defining vocabulary that indicates how often each of these words occurs with each other in LDOCE glosses. Each word can then be represented by a vector where each dimension shows often it occurs with another of the other words. The intuition here is that words that appear in similar contexts will be related in meaning. Given such information, a gloss can be expanded to include those other words that are related to the ones already used in the gloss. After a gloss is expanded, all of the word vectors are averaged into a single gloss vector that represents that particular sense. This is somewhat similar to our own gloss vector measure, although we do not expand the glosses with similar words and we rely on WordNet as our gloss corpus.

To perform word sense disambiguation, the context in which a target word occurs is also expanded to include words that are related to those already in the context. An averaged vector is created from all of the word vectors to represent the context, and this is the compared with the gloss vectors of the possible senses of the target word. The sense associated with the gloss vector that is most similar to the context of the target word is selected. Our method of disambiguation is distinct, in that Wilks, et. al. measures the relatedness of the target word's senses to the context, while we measure relative to the senses of the words in the context.

Cowie, et. al. [15] suggest that while the Lesk algorithm is capable (in theory) of disambiguating all the words in a sentence simultaneously, the computational complexity of such an undertaking is enormous and makes it difficult in practice. They employ simulated annealing to simultaneously search for the senses of all the content words in a sentence. If the assignment of senses was done using an exhaustive search the time involved would be prohibitive (since each possible combination of senses would have to be considered). However, simulated annealing can find a solution that globally optimizes the assignment

of senses among the words in the sentence without exhaustive search.

While quite a bit of research has been designed to extend and improve Lesk's algorithm, there has also been a body of work that is more directly linked to Quillian's spreading activation networks. For example, Veronis and Ide [16] represent the senses of words in a dictionary in a semantic network, where word nodes are connected to sense nodes that are then connected to the words that are used to define that sense. Disambiguation is performed via spreading activation, such that a word that appears in the context is assigned the sense associated with a node that is located in the most heavily activated part of the network.

Kozima and Furugori [17] construct a network from LDOCE glosses that consist of nodes representing the controlled vocabulary, and links to show the co-occurrence of these words in glosses. They define a measure based on spreading activation that results in a numeric similarity score between two concepts.

Niwa and Nitta [18] compare context vectors derived from co–occurrence statistics of large corpora with vectors derived from the path lengths in a network that represent their co–occurrence in dictionary definitions. In the latter case, they construct a Quillian–style network where words that occur together in a definition are linked, and those words are linked to the words that are used in their definitions, and so forth. They evaluate Wilk's et. al. context vector method of disambiguation, and find that dictionary content is a more suitable source of co–occurrence information than are other corpora.

2.2   Concept Hierarchies

The wide availability of WordNet as a concept hierarchy has led to the development of a number of approaches to disambiguation based on exploiting its structure.

Sussna [19] proposes a disambiguation algorithm assigns a sense to each noun in a window of context by minimizing a semantic distance function among their possible senses. While this is similar to our approach of disambiguation via maximizing relatedness, his disambiguation algorithm is based on a measure of relatedness among nouns that he introduces. This measure requires that weights be set on edges in the WordNet noun hierarchy, based on the type of relation the edge represents. His measure accounts for *is–a* relations, as well as *has–part, is–a–part–of*, and *antonyms*. This measure also takes into account the compressed edge lengths that exist at higher levels of the WordNet hierarchy, where a single link suggests a much greater conceptual distance than links lower in the hierarchy.

Agirre and Rigau [20] introduce a similarity measure based on *conceptual density* and apply it to the disambiguation of nouns. We refer to this as a measure of similarity since it is based on the *is–a* hierarchy in WordNet, and only applies to nouns. This measure is similar to the disambiguation technique proposed by Wilks, et. al. in that it measures the similarity between a target noun sense and the nouns in the surrounding context.

In order to perform disambiguation, Agirre and Rigau divide the WordNet noun *is–a* hierarchy into subhierarchies, where each possible sense of the ambiguous noun belongs to a subhierarchy. The conceptual density for each subhierarchy describes the amount of space occupied by the nouns that occur within the context of the ambiguous noun. In effect this measures the degree of similarity between the context and the possible senses of the word. For each possible sense the measure returns 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 is assigned to the target word.

Banerjee and Pedersen [1] suggest an adaptation of the original Lesk algorithm in order to take advantage of the network of relations provided in WordNet. Rather than simply considering the glosses of the surrounding words in the sentence, the concept network of WordNet is exploited to allow for glosses of word senses related to the words in the context to be compared as well. In effect, the glosses of surrounding words in the text are expanded to include glosses of those words to which they are related through relations in WordNet. Pedersen and Banerjee also suggest a scoring scheme such that a match of n consecutive words in two glosses is weighted more heavily than a set of n one word matches. The work in this article represents a considerable refinement to this earlier work, and there are a number of substantial changes both to the disambiguation algorithm and the measures of relatedness that are employed.

3 Maximum Relatedness Disambiguation

This article introduces an algorithm that uses measures of semantic relatedness to perform word sense disambiguation. This algorithm finds its roots in the original Lesk algorithm, which disambiguates a polysemous word by picking that sense of the target word whose definition has the most words in common with the definitions of other words in a given window of context. Lesk's intuition was that related word senses will be defined using similar words, and there will be overlaps in their definitions that will indicate their relatedness. We generalize this approach by creating an algorithm that can perform disambiguation using any measure that returns a relatedness or similarity score for pairs of word senses.

7

We denote the words in a window of context as w_1, w_2,..., w_n, where w_t, $1 \leq t \leq n$, is the target word to which a sense must be assigned. Assume that each word w_i has m_i possible senses, denoted as s_{i1}, s_{i2}, ..., s_{im_i}. The goal of any disambiguation algorithm is to select one of the senses from the set $\{s_{t1}, s_{t2}, ..., s_{tm_t}\}$ as the most appropriate sense for the target word w_t.

Our algorithm performs word sense disambiguation by using a measure of semantic relatedness that is denoted as $relatedness : s_{ij} \times s_{kl} \rightarrow R$, where s_{ij} and s_{kl} represent any two senses of the words in the window of context, and R represents the set of real numbers. In other words, $relatedness$ is a function that takes as input two senses, and outputs a real number. Further, the algorithm assumes that this output number is indicative of the degree of semantic relatedness between the two input senses. In the Lesk algorithm for instance, the $relatedness$ function would return the number of words that overlap between the definitions of the two input senses, and the larger this number, the more related the two senses are.

Using this notation, we can concisely describe our word sense disambiguation algorithm using equation 1.

$$argmax_{i=1}^{m_t} \sum_{j=1,j \neq t}^{n} max_{k=1}^{m_j} relatedness(s_{ti}, s_{jk}) \tag{1}$$

This equation shows that the algorithm computes a score for each sense s_{ti} of the target word. The output of the algorithm is the index of the sense of the target word that is most related to the other words in the window of context, and is therefor considered to be the most appropriate sense.

The algorithm is also described in Figure 1. For every word w_j in the window of context, the algorithm computes the relatedness between s_{ti} and each sense s_{jk}, $k \leq 1 \leq m_j$, of word w_j and picks the highest such relatedness score. The algorithm then adds the score from each of the words in the window, and this becomes the score for sense $s_t i$ of the target word. That sense with the highest such score is returned as the appropriate sense of the target word. Note that it is possible for more than one sense of the target word to have the same highest score; in this case we report all such tied senses instead of attempting to choose a single sense from among them.

For example, consider the sentence *I put the check in the bank*. As described in section 4, WordNet does not store information on pronouns, prepositions or determiners, so the only words that have senses in WordNet are *put, check* and *bank*. For simplicity's sake, assume that the word *put* has one sense in WordNet denoted by put_1: *put into a certain place or abstract location*, the word *check* has two senses, $check_1$: *a written order directing a bank to pay money*, and

```
foreach sense $s_{ti}$ of target word $w_t$

{
    set $score_i = 0$
    foreach word $w_j$ in window of context
    {
        skip to next word if j == t
        foreach sense $s_{jk}$ of $w_j$
        {
            temp_score[j] = relatedness($s_{ti}$, $s_{jk}$)
        }
        winning_score = highest score in array temp_score[]
        if (winning score > threshold)
        set $score_i = score_i +$ winning_score
    }
}
return i, such that $score_i \geq score_j$, $\forall j, 1 \leq j \leq n$, $n =$
number of words in sentence
```

Fig. 1. Pseudo Code of Maximum Relatedness Disambiguation

$check_2$: *the bill in a restaurant*, and that the word *bank* has two senses: $bank_1$: *financial institution that accepts deposits and channels the money into lending activities*, and $bank_2$: *sloping land especially beside a body of water*. Finally assume that the target word is *check*.

Our disambiguation algorithm takes as input such a sentence, and also a relatedness measure m that takes as input two synsets (like $check_1$ and $bank_2$ for example) and outputs a real number that is proportional to the degree of relatedness between the two input synsets. Given such a measure, the algorithm computes a score for $check_1$ as follows: For each neighboring word, the algorithm uses m to get the relatedness scores between $check_1$ and each sense of the word, and then picks the highest of these scores. For the word *put*, this amounts to the single score $m(check_1, put_1)$, while for the word *bank*, this amounts to taking the greater of the scores $m(check_1, bank_1)$ and $m(check_1, bank_2)$. The algorithm then adds these scores to arrive at the overall score for $check_1$. That is, score($check_1$) $= m(check_1, put_1) + MAX(m(check_1, bank_1), m(check_1, bank_2))$. Similarly, a score is obtained for $check_2$: score($check_2$) $= m(check_2, put_1) + MAX(m(check_2, bank_1), m(check_2, bank_2))$. Finally, the algorithm compares score($check_1$) and score($check_2$) and reports that sense that has a higher score.

We can apply this algorithm using any measure of semantic relatedness, regardless of how it is computed or what it is based on. In this article, we carry out disambiguation relative to WordNet senses, so we briefly introduce Word-Net before going on to describe the various measures of relatedness that we

employ.

4 WordNet

Due to its increasing scope and free availability, WordNet has become a popular resource for identifying taxonomic and networked relationships among concepts. Since it also includes glosses for word senses, it is also often used as a machine readable dictionary, although the WordNet team prefers that it be known as a lexical database.

WordNet [21] contains information about nouns, verbs, adjectives and adverbs. It organizes related concepts into *synonym sets* or *synsets*. Each synset can be thought of representing a *concept* or *word sense*. For example: {*car*, *auto*, *automobile*, *machine*, *motorcar*} is a synset that represents the sense defined by the gloss: *4–wheeled motor vehicle; usually propelled by an internal combustion engine.* In effect each synset represents a concept or word sense, and we will use these terms somewhat interchangeably.

In addition to providing these groups of synonyms to represent a concept, WordNet connects concepts via a variety of relations. This creates a network where related concepts can be (to some extent) identified by their relative distance from each other. The relations provided include *synonymy, antonymy, is–a*, and *part–of.*

Relations in WordNet generally do not cross part of speech boundaries, so semantic and lexical relations tend to be between concepts with the same part of speech. However, with the release of WordNet 2.0 in the summer of 2003, there are now links between derived forms of noun and verb concepts, and there are also domain relations that include noun and verb concepts. The increased interconnectivity of WordNet will offer interesting opportunities for future work, although as of this writing we have not yet taken advantage of this.

For nouns the most common and useful relation is the *is–a* relation. This exists between two concepts when one concept *is–a–kind–of* another concept. Such a concept is also known as a *hypernym*. For example, a *car* is a hypernym of *motor vehicle.*

The *is–a* hierarchy of noun concepts is perhaps the distinguishing characteristic of WordNet. These comprise over 70% of the total relations for nouns. An *is–a* hierarchy also exists for verbs, although it represents *is–way–of–doing*, also known as *troponomy*. As an example, *walking* is a troponym of *moving.* Each hierarchy (be it for nouns or verbs) can be visualized as a tree that has

a very general concept associated with a root node and more specific concepts associated with leaves. For example, a root node might represent a concept like *entity* whereas leaf nodes are associated with *carving fork* and *whisk broom.*

We use WordNet 1.7 which contains nine separate noun hierarchies containing 74,588 concepts joined by 76,226 *is–a* links. In order to allow for paths between all noun concepts in WordNet, we create an artificial root node that subsumes the nine given hierarchies. The verb hierarchies provide less information about similarity between concepts since there are 628 separate hierarchies for the 12,754 verb sense. While these could also be joined by a single root node, the result would be a tree structure that was very wide and shallow, since most of these hierarchies have between two to five levels. As a result it would be hard to differentiate among concepts connected only via that artificial root. Adjectives are not arranged in a hierarchy, so the issue of having a subsuming root node does not apply.

5   Measures of Relatedness and Similarity

Thus far we have used the term semantic relatedness fairly freely, and have sometimes mentioned semantic similarity as well. Before we discuss the various measures we have studied in detail, we should clarify the distinction between these two terms.

Two concepts can be related without being similar, so relatedness should be seen as a more general notion than similarity. For example, two concepts may be related because they are antonyms, but they are not likely to be considered similar.

We use the term similarity in a very specific sense, that is it refers to a relationship between concepts that is based on information as found in an *is–a* hierarchy. In the case of WordNet, this limits similarity judgments to be between pairs of nouns or pairs of verbs, since the concept hierarchies in WordNet do not mix parts of speech. As a practical matter, only the noun hierarchies are extensive enough to allow for relatively fine grained distinctions among related concepts.

5.1   *Path Based Measures*

When given an *is–a* hierarchy, one means of determining the degree to which two concepts are related is to count the number of edges between them, or to find the length of shortest path between two concepts.

In principle path based measures can apply to any taxonomy. Thus, in our experimental evaluation we attempted to employ path length measures (and information content measures, that will be described shortly) with both nouns and verbs. In many cases these did not fare well with verbs, which is to be expected since the verb hierarchies in WordNet are shallow and plentiful. As a result very few verb concepts actually occupy the same hierarchy and there will rarely be paths between verb concepts. However, it's important to note that this reflects more upon a limitation in WordNet than something inherent in these measures.

Unfortunately, path lengths are most appropriate when they have a relatively consistent interpretation throughout the taxonomy or network. This is not the case with WordNet, since concepts higher in a hierarchy are more general than those lower in the hierarchy. Thus, a path of length one between two general concepts can suggest a large difference whereas one between two specific concepts may not. For example, *mouse* and *rodent* are separated by a path of length one, which is the same distance that separates *fire iron* and *implement*.

The fact that path lengths can be interpreted differently depending on where they occur in WordNet has led to the development of a number of measures based on path lengths that incorporate a variety of correcting factors.

5.1.1 Rada, et. al.

Rada, et. al. [22] define the conceptual distance between any two concepts as the the shortest path through a semantic network. They evaluate this technique using MeSH, a hierarchical semantic network of biomedical concepts that (at that time) consisted of about 15,000 terms organized into a nine-level hierarchy. This measure is similar in spirit to approaches that rely on spreading activation, and works relatively well due to that the fact that the network consists of concepts consists of *broader-than* relationships, which includes both *is–a* and *part-of* relationships. In this technique, the number of edges in the shortest path between two concepts under consideration gives the measure of similarity.

5.1.2 Leacock and Chodorow

The measure of Leacock and Chodorow [6] is related to that of Rada, et. al., in that it is based on the length of the shortest paths between noun concepts in an *is–a* hierarchy. The shortest path is the one which includes the fewest number of intermediate concepts. This value is scaled by the depth D of the hierarchy, where depth is defined as the length of the longest path from a leaf node to the root node of the hierarchy.

Thus, their measure of similarity is defined as follows:

$$sim_{lch}(c_1, c_2) = max[-log(length(c_1, c_2)/(2 \cdot D))] \qquad (2)$$

where $length(c_1, c_2)$ is the shortest path length (i.e., having minimum number of nodes) between the two concepts and D is the maximum depth of the taxonomy. Given that we introduce a hypothetical root node in WordNet that joins all the noun hierarchies, D becomes a constant of 16 for all noun concepts, meaning that the path length from this root node to the most distant leaf is 16 in WordNet 1.7.

5.1.3 Wu and Palmer

Wu and Palmer [5] define a measure of similarity that is also based on path lengths, however, they focus on the distance between a concept to the root node.

Resnik [23] reformulates their measure slightly, and we follow that presentation here. This measure finds the distance to the root of the most specific node that intersects the path of the two concepts in the *is–a* hierarchy. This intersecting concept is the most specific concept that the two concepts have in common, and is known as the lowest common subsumer (lcs). The distance of the lcs is then scaled by the sum of the distances of the individual concepts to the node. The measure is formulated as follows:

$$sim_{wup}(c_1, c_2) = \frac{2 * depth(lcs(c_1, c_2))}{depth(c_1) + depth(c_2)} \qquad (3)$$

where $depth$ is the distance from the concept node to the root of the hierarchy. As will become apparent shortly, this measure can be thought of as a path based equivalent of the Lin similarity measure. It is interesting to note that Wu and Palmer describe this measure relative to a verb taxonomy, but in fact it applies equally well to any part of speech as long as the concepts are arranged in a hierarchy.

5.1.4 Hirst and St.Onge

Hirst and St. Onge [7] introduce a measure of relatedness that considers many other relations in WordNet beyond *is–a* relations. This measure is unique among those discussed thus far, in that those have all been measures of similarity and focus on *is–a* hierarchies. The effect of this is that the measure is able to assess the relatedness between heterogeneous pairs of parts of speech. For example, it can determine the relatedness between a noun and a verb.

None of the other path based measures or the information content measures that will be discussed have this capability.

This measure was originally used to identify lexical chains, which are a series of related words that maintain coherence in a written text. Since it was originally intended to find relations among words (and not concepts) we have made a few adaptations to the measure as originally described.

This measure classifies all WordNet relations as horizontal, upward, or downward. Upward relations connect more specific concepts to more general ones, while downward relations join more general concepts to more specific ones. For example, *is–a* is an upward relation while *is–a–kind–of* is considered to be a downward relation. Horizontal relations (such as *antonyms*) maintain the same level of specificity.

The Hirst–St. Onge measure has four levels of relatedness: extra strong, strong, medium strong, and weak. An extra strong relation is based on the surface form of the words and therefore does not apply in our case since we are measuring the relatedness of word senses.

Two words representing the same concept (e.g., synonyms) have a strong relation between them. Thus, there is a strong relation between two instances of the same concept. There are two additional scenarios by which a strong relations can exist. First, if the synsets representing the concepts are connected via a horizontal relation, as in the case of opposites joined by an antonym relation. Second, if one of the concepts is represented by a compound word and the other concept is represented by a word which is a part of the compound, and if there is any kind of synset relation between the two concepts. For example, *racing_car* and *car* are considered to have a strong relation, since *car* occurs in both, and they are joined via an *is–a* relation.

The medium–strong relation is determined by a set of allowable paths between concepts. If a path that is neither too long nor too winding exists, then there is a medium–strong relation between the concepts. The score given to a medium–strong relation considers the path length between the concepts and the number of changes in direction of the path:

$$path\_weight = C - path\_length - (k \times \#\_changes\_in\_direction) \qquad (4)$$

Following Budanitsky and Hirst [24], we set C to 8 and k to 1. The value of strong relations is defined to be 2 * C. Thus, two concepts that exhibit a strong relation will receive a score of 16, while two concepts with a medium–strong relation will have a maximum score of 8, and two concepts that have no relation will receive a score of zero.

Information content [3] is a measure of specificity that is assigned to each concept in a hierarchy. A concept with a high information content is very specific to a particular topic, while concepts with lower information content are associated with more general, less specific concepts. Thus, *carving fork* has a high information content while *entity* has low information content.

Information content of a concept is estimated by counting the frequency of that concept in a large corpus and thereby determining its probability via a maximum likelihood estimate.The information content of a concept is defined as the negative log probability of the concept.

$$IC(concept) = -log(P(concept)) \tag{5}$$

The frequency of a concept includes the frequency of all its subordinate concepts since the count we add to a concept is added to its subsuming concept as well. Note that the counts of more specific concepts are added to the more general concepts, but not from the more general to specific. Thus, counts of more specific concepts percolate up to the top of the hierarchy, incrementing the counts of the more general concepts as they proceed upward. As a result, concepts that are higher up in the hierarchy will have higher counts than those at lower more specific levels and have higher probabilities associated with them. Such high probability concepts will have low values of information content since they are associated with more general concepts.

If sense–tagged text is available, frequency counts of concepts can be attained directly, since each concept will be associated with a unique sense. If sense–tagged text is not available it will be necessary to adopt an alternative counting scheme. Resnik [25] suggests counting the number of occurrences of a word type in a corpus, and then dividing that count by the number of different concepts/senses associated with that word.

We have computed information content using SemCor and the British National Corpus (BNC). SemCor is a 200,000 word sense–tagged sample of text, about 80% of which comes from the Brown Corpus and the remaining 20% comes from the novel *The Red Badge of Courage*. Since the sense tags are from WordNet, the concept counts can be taken directly from the sense tagged text. BNC is a 100,000,000 word sample of modern British English that is derived from a variety of sources. It has not been sense tagged, and as such we adopt Resnik's method and distribute concept counts across the possible senses of a word.

5.2.1 Resnik

The Resnik measure of semantic similarity [3] is based on the information content of noun concepts as found in the *is–a* hierarchies of WordNet. The principle idea behind this measure is that two concepts are semantically related proportional to the amount of information they share in common. The quantity of information common to two concepts is determined by the information content of their lowest common subsumer. Thus, the Resnik measure of similarity is defined as follows:

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

We note that this is does not consider the information content of the concepts being measured, nor does it directly consider the path length between them. The potential limitation that this poses is that quite a few concepts might share the same least common subsumer, and will have identical values of similarity assigned to them. For example, in WordNet the concept of *vehicle* is the least common subsumer of *jumbo jet, tank, house trailer*, and *ballistic missile*. Therefore any pair of these concepts would receive the same similarity score. This is particularly troublesome with verbs in WordNet, since there are a large number of verb hierarchies, and a pair of verb concepts are not likely to have any lowest common subsumer, or they are subsumed by some hypothetical root note that we introduce to link together all of the verb hierarchies. Thus, the Resnik measure might best be considered as coarse grained measure, and subsequent measures have attempted to refine it and give it greater ability to distinguish similarity among concepts.

5.2.2 Jiang and Conrath

Jiang and Conrath [8] define a measure of semantic distance for nouns that relies on Resnik's measure. The intuition behind the measure is that the difference between the information content of the individual concepts and that of their lowest common subsumer will reveal how similar or different they are. If the sum of their individual information contents is close to that of their lowest subsumer, then it suggests that the measures are located close together in the concept hierarchy. Thus, they take the sum of the information content of the individual concepts and subtract from that the information content of their lowest common subsumer:

$$dist_{jcn}(c_1, c_2) = IC(c_1) + IC(c_2) - 2 \times IC(lcs(c_1, c_2)) \tag{7}$$

Since this is a distance measure, concepts that are more similar have a lower score than the less similar ones. In order to maintain consistency among the

measures, we convert this measure to semantic similarity by taking its inverse: following:

$$sim_{jcn}(c_1, c_2) = \frac{1}{dist_{jcn}(c_1, c_2)} \qquad (8)$$

5.2.3 Lin

The Lin measure [9] of similarity measures the ratio of the information content needed to state the commonality of the two concepts as represented in their lowest common subsumer to the amount of information needed to describe them individually.

The commonality of two concepts is captured by the information content of their lowest common subsumer and the information content of the two concepts themselves. This measure turns out to be a close cousin of the Jiang–Conrath measure, although they were developed independently:

$$sim_{lin}(c_1, c_2) = \frac{2 \times IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} \qquad (9)$$

Lin points out that this measure is related to the well–known Dice Coefficient, and that the measure of Wu and Palmer can be thought of as a special case of the Lin measure.

5.3 Gloss Based Measures

We believe that gloss overlaps are a very promising means of measuring relatedness, since they can be used to make comparisons between concepts of different parts of speech. For example, this might include comparing nouns with verbs, or verbs with adjectives. Measures that are based on paths in *is–a* hierarchies tend to be limited to making comparisons between concepts with the same part of speech, since these hierarchies do not include multiple parts of speech. The only other measure capable of mixed part of speech comparisons is that of Hirst and St. Onge, which is dependent on the existence of specific links between concepts.

However, we do recognize that glosses are by necessity short, and may not provide sufficient information on their own to make judgments about relatedness. For example, the gloss of *canoe* is *small and light boat pointed at both ends propelled with a paddle*. It has no gloss overlaps with either *bank_1*: *financial institution that accepts deposits and channels the money into lending activities*, or with *bank_2*: *sloping land especially beside a body of water*. Thus

in the sentence *The canoe was near the bank*, a simple gloss overlap measure finds no relation between *canoe* and either sense of *bank*.

We have developed two different measures to address this issue. In the extended gloss overlap measure, we also make comparisons between glosses of words that are related according to WordNet. In the gloss vector measure simplify ideas from both Wilks, et. al. [14] and Schütze [26] to create a relatedness measure based on dictionary gloss co–occurrence statistics.

5.3.1 Extended Gloss Overlaps

The extended gloss overlap measure was developed to overcome the limitations of short definitions [10]. Lesk's gloss overlaps are adapted to a networked resource such as WordNet by finding overlaps not only between the definitions of the two concepts being measured, but also among those concepts to which they are related.

This is motivated by the idea that semantic relations (such as *is–a* and *has–part*) specified in WordNet do not capture all the possible relations between concepts. For example, there are no explicit relations between *boat: a small vessel for travel on water*, and $bank_2$: *sloping land especially beside a body of water*. Even the shortest *is–a* path between them in WordNet is not particularly indicative of their relatedness, since it includes the higher level concept *physical object*.

However, despite the lack of a path in WordNet (direct or indirect) we observe that *boat* and $bank_2$ are related. One can launch a boat from a bank, for example, or run a boat aground on a bank. The glosses of these two concepts share the word *water* which hints at their relatedness. The fact that concepts to which each of these are related also share overlaps adds to that conclusion. Thus, in general we believe that there are relations between concepts that are implicit but can be found via gloss overlaps.

For the extended gloss overlap measure, we consider the glosses of all the concepts that are directly connected to a concept by a relation when finding overlaps. The process of finding and scoring overlaps can be described as follows: When comparing two glosses, we define an overlap between them to be the longest sequence of one or more consecutive words that occurs in both glosses such that neither the first nor the last word is a function word, that is a pronoun, preposition, article or conjunction. If two or more such overlaps have the same longest length, then the overlap that occurs earliest in the first string being compared is reported. Given two strings, the longest overlap between them is detected, removed and in its place a unique marker is placed in each of the two input strings. The two strings thus obtained are then again checked for overlaps, and this process continues until there are no longer any overlaps

between them. The sizes of the overlaps thus found are squared and added together to arrive at the score for the given pair of glosses.

The original Lesk Algorithm compares the glosses of a pair of concepts and computes a score by counting the number of words that are shared between them. This scoring mechanism does not differentiate between single word and phrasal overlaps and effectively treats each gloss as a bag of words. For example, it assigns a score of 3 to $bank_2$: (sloping land especially beside a body of water) and lake: (body of water surrounded by land), since there are 3 overlapping words: land, body, water. Note that stop words are removed, so of is not considered an overlap.

However, there is a Zipfian relationship [27] between the lengths of phrases and their frequencies in a large corpus of text. The longer the phrase, the less likely it is to occur multiple times in a given corpus. A phrasal n–word overlap is a much rarer occurrence than a single word overlap. Therefore, we assign an n word overlap the score of n^2. This gives an n–word overlap a score that is greater than the sum of the scores assigned to those n words if they had occurred in two or more phrases, each less than n words long. This is true since the square of a sum of positive integers is strictly greater than the sum of their squares. That is, $(a_0 + a_1 + ... + a_n)^2 > a_0^2 + a_1^2 + ... + a_n^2$, where a_i is a positive integer. For the above gloss pair, we assign the overlap land a score of 1 and body of water a score of 9, leading to a total score of 10.

5.4   Gloss Vectors

Even with extended gloss overlaps we are still concerned that in some cases the glosses may not contain enough overlaps to make fine grained relatedness decisions. As a result, we have developed a method that represents concepts as gloss vectors, and measures the similarity of concepts by finding the cosine between their respective gloss vectors [11].

Our measure can be fairly viewed as a simplification of Wilks, et. al. [14], since we build word vectors from gloss co–occurrence data, and then build a gloss vector from the average of those. However, we do not not expand the glosses with similar words but simply use those that appear in the gloss. In fact, we have the capability now to expand the glosses with those of related concepts, like we do in the extended gloss overlap measure, but those results are not reported here.

It must be pointed out as well that we draw considerable motivation and insight from observations made by Schütze [26]. He describes a similar process of creating context vectors from word vectors, in fact he represents the context as an average of these word vectors. However, his work is distinct in that he is

using large corpora that is not dictionary text, and he employs Singular Value Decomposition to reduce the dimensionality of his word vectors. Rather than comparing his context vectors to a vector that represents a dictionary sense, he simply clusters those context vectors in order to discover senses without regard to any existing inventory.

However, the significant contribution Schütze makes to our work is his explanation of what context vectors are capturing, which revolves around the notion of a second order co–occurrence. This is an indirect relationship between a pair of words, in that these are words that do not occur together but both occur with some third word. For example, in *product line* and *telephone line*, *product* and *telephone* are first order co–occurrences with *line*, and they are second order co–occurrences with each other by virtue of this first order relationship. Schütze argues that second order relationships are more representative of meaning, and tend to be less sparse than first order co–occurrences.

Our gloss vector measure treats the WordNet (version 1.7) glosses as a 1,400,000 word corpus of plain text. The first step in deriving gloss vectors is to build a co–occurrence matrix of the words that occur in the corpus. This matrix represents the number of time any two words occur together in a WordNet gloss. Note that we eliminate certain non-content stop words, as well as words that occur more than 1,000 times and less than 5 times, which reduces the size of the corpus to about 1,200,000 words. The resulting matrix is approximately 15,000 x 15,000 and is both symmetric and relatively sparse. Each cell tells the number of times the words represented by the row and the column occur together in a WordNet gloss.

To measure the relatedness of a pair of concepts, a vector is constructed for each of the glosses. First, the row (or column) entry for each word in the gloss is found in the co–occurrence matrix, and the entire row is treated as a vector. This will show the number of times that word has occurred in a gloss with all of the other words that have appeared in WordNet glosses (and have fallen above and below our frequency cutoffs). After all the words in the gloss are represented with a vector, we find the average of all these word vectors, and use this single vector to represent the meaning of the concept. In effect, it is a second order co–occurrence representation of the words that co–occur with the words in the gloss.

After the gloss vectors are created for each concept, we compare pairs of concepts by measuring the cosine of the angle between their corresponding gloss vectors. Note that two concepts are considered to be related when they share a set of words that co-occur with the words that occur in their respective glosses. The idea here is that gloss overlaps may be somewhat unlikely to find because they are so short. However, if we consider the words that co–occur with the words in the glosses, this becomes a larger set of a words that can

be matched in a more refined manner.

For example, the gloss of *lamp* is *an artificial source of visible illumination*. The gloss vector for *lamp* is created by finding the average of the word vectors of *artificial*, *source*, *visible* and *illumination*. Suppose that this is being compared with *sun*, which has the gloss *a typical star that is the source of light and heat for the planets in the solar system*. While there is indeed a first order co–occurrence overlap of *source*, that is fairly limited evidence upon which to measure relatedness. However, in the WordNet gloss corpus (or any corpus of text) *illumination* and *light* are likely to be used with a similar set of words, and their commonality will be captured in their corresponding word vectors.

In fact we could follow Schütze and create the underlying word co–occurrence matrix from any corpus. While this remains an avenue for future work, at present we have focused on only using the WordNet gloss corpus since we are interested in seeing how well dictionary content alone will fare. In particular, we want to see how well gloss vectors based on second–order co–occurrences compare with the more traditional first order co–occurrences as used by the Lesk algorithm and in the extended gloss overlap measure.

6 Experimental Data

We evaluate our algorithm using the Senseval-2 English lexical sample data (test portion) [28]. This consists of 4,328 *instances* each of which contains a sentence with a single target word to be disambiguated, and one or two surrounding sentences that provide additional context. There is a gold standard tagging that was created by human annotators, and we use this only to evaluate the results of our algorithm.

There are 73 different target words in the sample: 29 nouns, 29 verbs, and 15 adjectives. Table 1 shows the words, their frequency count (or number of instances), and the number of WordNet senses in which they are used according to the gold standard data. It should be noted that the degree of difficulty of this data (as judged by the number of possible senses) is relatively high. For the 29 nouns, on average they are used in 8.2 sense. The 29 verbs are used in an average of 12.2 senses, and the 15 adjectives are used in an average of 7.1 senses.

Disambiguation is carried out by measuring the relatedness between the senses of each word in the window of context and the possible senses of the target word. For every instance, the window is defined such that the target word is at the center (if possible). We experiment with windows of six different sizes: 2, 3, 5, 11, 21, and 51. The size of the window includes the target word, and

Table 1
Experimental Data

Nouns	Count	Senses	Verbs	Count	Senses	Adjectives	Count	Senses
art	98	15	begin	280	7	blind	55	6
authority	92	8	call	66	17	colourless	35	3
bar	151	18	carry	66	20	cool	52	7
bum	45	6	collaborate	30	2	faithful	23	3
chair	69	8	develop	69	14	fine	70	14
channel	73	11	draw	41	22	fit	29	3
child	64	5	dress	59	12	free	82	13
church	64	5	drift	32	9	graceful	29	2
circuit	85	16	drive	42	13	green	94	14
day	145	12	face	93	6	local	38	4
detention	32	7	ferret	1	1	natural	103	23
dyke	28	4	find	68	17	oblique	29	3
facility	58	5	keep	67	20	simple	66	5
fatigue	43	6	leave	66	10	solemn	25	2
feeling	51	6	live	67	9	vital	38	4
grip	51	7	match	42	7			
hearth	32	5	play	66	20			
holiday	31	5	pull	60	25			
lady	53	8	replace	45	4			
material	69	10	see	69	13			
mouth	60	11	serve	51	11			
nation	37	4	strike	54	20			
nature	46	7	train	63	8			
post	79	10	treat	44	5			
restraint	45	9	turn	67	26			
sense	53	12	use	76	6			
spade	33	6	wander	50	5			
stress	39	7	wash	12	7			
yew	28	4	work	60	18			
Total:	1754	(avg.) 8.2	Total:	1806	(avg.) 12.2	Total:	768	(avg.) 7.1

the number of words both to the right and left. The window size of 2 indicates that it includes the target word and one word to the left. A window size of 3 includes the target word and one word to the left and right, and so forth.

The window of context may include more than one sentence, since in the Senseval-2 data most of the instances are made up of 2-3 sentences. The window of context consists of words that are known to WordNet. This has the effect of eliminating quite a few function (stop) words, but a stop list is still needed to specify words that should be excluded from the window. This is because some some function and low content words happen to have an unusual or infrequently used WordNet sense and would thereby be included in the window of context. For example, *who* is known to WordNet as an abbreviation for the World Health Organization, and *in* has three nouns senses: an abbreviation for *inches*, the element Indium and the state of Indiana.

We part of speech tagged the Senseval-2 data, but found no particular improvement in the accuracy of disambiguation when restricting the senses of a word in the window to those that belonging to the designated part of speech. Given the minimal impact this had on the quality of results, we do not use part of speech information about the words in the window of context. When using the path based measures and the information content measures, we only consider the noun senses of these words, regardless of the actual part of speech in which they are used. When using hso and the gloss based measures, we consider all the possible senses for all the possible parts of speech of a word. In the Senseval-2 data, each target word is used in only one part of speech, and this information is provided in the test data as used in that event. Therefore we also use this information, so the part of speech of the target word is restricted to what is intended in that context.

The fact that we don't use part of speech tags nor sentence boundary information may raise the concern that we introduce too much noise into the process. However, an effective measure of relatedness will score unrelated senses very low, and they will be overwhelmed by the higher scores attained by the more related senses. Thus, the measure of relatedness can act as its own filter and remove the noise that unrelated senses bring, whether they be caused by part of speech differences, sentence boundary overruns, etc.

7 Experimental Methodology

In order to evaluate our algorithm, we conducted an extensive set of experiments using nine different measures of relatedness. In addition, there were two baseline measures included. One simply generates random relatedness values, rather than computing them in some principled way. The other employs a sim-

ple edge counting method that treats the length of the shortest path between two concepts as the relatedness value. The random baseline serves as a sanity check, and attains the accuracy that would be expected by randomly guessing from the set of possible senses for each instance. The edge measure represents the simplest and most intuitive approach, and is a useful point of comparison in that several of the measures we use are intended to correct problems with simple edge counting.

The measures and the abbreviations by which we will often refer to them are summarized below.

- Baselines
  · Random [random] : guess from set of possible senses
  · Path Length [edge] : find shortest path between concepts in *is–a* hierarchy

- Path Based Measures
  · Wu and Palmer [wup] : distance to root in *is–a* hierarchy, lin without information content
  · Leacock and Chodorow [lch] : shortest *is–a* path between concepts, scaled by depth of taxonomy
  · Hirst and St. Onge [hso] : upward, downward, and horizontal paths using many relations, can compare across parts of speech

- Information Content Measures
  · Resnik [res] : Information Content (IC) of shared concept
  · Lin [lin] : IC of shared concept scaled by individual concept ICs
  · Jiang and Conrath [jcn] : sum of individual ICs minus shared IC

- Gloss Based Measures (all can compare across parts of speech)
  · Original Lesk [lesk-o] : find gloss overlaps between two concepts
  · Extended Gloss Overlaps [lesk-e] : find gloss overlaps of two concepts, plus those to which they are connected in WordNet
  · Gloss Vector [vector] : represent a concept as an averaged vector of words vectors derived from gloss co–occurrence data

8 Experimental Results

We evaluate our algorithm by comparing its results with the human created gold standard. We compute *precision*, which is the number of correct answers divided by the number of answers reported by a system, and *recall*, the number of correct answers divided by the number of instances in the same. We report a summary of these two values known as the *F–measure*, which is the harmonic

24

mean of the precision and recall and is formulated as follows:

$$F - measure = \frac{2 * precision * recall}{(precision + recall)} \qquad (10)$$

We use the key and scoring software exactly as provided by the Senseval-2 organizers, and all results we report are based on fine grained scoring, which requires an exact match between the system output and the manually specified answer in the gold standard.

It is possible there there be a tie for the most related target word sense, and in that case all those senses are reported as answers and partial credit is given if one of them prove to be correct. This can reasonably occur if a word is truly ambiguous, or if the meanings are very closely related and it is not possible to distinguish among them. It is also possible that none of the target word senses will receive a score greater than zero. In this case, no answer is reported by the algorithm since there is no evidence to choose one sense over another.

We report the results of our experiments first by part of speech, and then across all parts of speech.

8.1 Nouns

Table 2 shows the F–measure of each of the measures when applied to disambiguating the 1,754 instances of the 29 nouns. All of the measures are well defined in relation to nouns, so this represents a fair comparison amongst all the measures.

The extended gloss overlap measure (lesk-e) and jcn-sem attain the highest F–measures for nouns. lesk-e reach a high score of .412, while jcn-sem attained .401. It should be noted that lesk-e was the most accurate measure for every window size, and in many cases by a large margin. This suggests that the content of noun glosses is particularly good, and the distinctions that they draw are fairly clear.

It is interesting to note that jcn-bnc did not fare nearly as well as jcn-sem, suggesting that the sense–tagged text was in fact helpful to this measure. While this is not surprising, what is curious is that the Lin measure (lin) performed at nearly the same levels with and without sense tagged text. It remains an interesting issue to determine why these two seemingly similar measures fare rather differently at this task.

A number of measures perform at levels less than random guessing when given small windows of context. This is due to a large number of unattempted

Table 2
F–measure (1,754 instances of 29 nouns)

Measure	Window Size					
	2	3	5	11	21	51
random	.200	.186	.193	.223	.209	.202
edge	.197	.223	.230	.242	.227	.227
lch	.197	.220	.226	.237	.226	.200
wup	.196	.244	.272	.288	.300	.276
hso	.123	.145	.206	.208	.197	.188
res-sem	.184	.229	.260	.285	.291	.291
res-bnc	.184	.230	.268	.293	.295	.274
lin-sem	.190	.229	.265	.290	.292	.283
lin-bnc	.212	.259	.287	.314	.326	.317
jcn-sem	.127	.330	.364	.386	.401	.358
jcn-bnc	.260	.282	.305	.332	.351	.320
lesk-o	.130	.200	.269	.280	.281	.269
lesk-e	**.326**	**.377**	**.396**	**.405**	**.412**	**.384**
vector	.284	.296	.289	.303	.292	.281

instances for these measures, where no relation was found between the target word senses and its immediate 1 or 2 neighbors. However, not all measures are susceptible to this; note that even for the smallest window of context, the extended gloss overlap measure (lesk-e) does extremely well. This shows that it is quite robust even when given a limited number of words in the window of context.

We also note that the measures generally increase in accuracy as the size of the window increases. This suggests that the method of disambiguation is relatively resistant to noise, and that increasing context provides better information to make a judgment about the relatedness of the target word to the text in which it occurs.

8.2   Verbs

The F–measure results for verbs are shown in Table 3. It must be pointed out that few of these measures actually advertise themselves as being suitable for verbs. For example, information content and path based measures have

Table 3
F–measure (1,806 instances of 29 verbs)

	Window Size					
Measure	2	3	5	11	21	51
random	.101	.099	.101	.110	.081	.104
edge	.093	.124	.139	.151	.158	.143
lch	.093	.112	.125	.131	.142	.133
wup	.024	.036	.040	.045	.044	.039
hso	.037	.051	.078	.088	.053	.062
res-sem	.025	.043	.052	.054	.039	.032
res-bnc	.025	.039	.047	.050	.050	.043
lin-sem	.028	.046	.055	.064	.058	.052
lin-bnc	.029	.046	.055	.061	.062	.058
jcn-sem	.127	.183	.195	.190	.197	.189
jcn-bnc	.084	.107	.107	.115	.121	.121
lesk-o	.070	.100	.119	.131	.134	.133
lesk-e	**.154**	**.198**	**.202**	**.212**	**.201**	**.195**
vector	.147	.163	.169	.178	.189	.167

generally been applied to noun concepts. However, in theory it is at least possible to compute information content for verbs in WordNet, since they are arranged in concept hierarchies (albeit small) that can be used to propagate counts, and paths can indeed be found between concepts. To some extent the inclusion of these measures in the verb experiment is speculative, and meant to test their limits.

While the results are generally rather low, there are a few encouraging signs. The gloss based measures and jcn performed at levels greater than random guessing, which for verbs is non–trivial given the large number of possible senses that exist for each target word on average. The extended gloss overlap measure again performed at the highest level for all window sizes, confirming that it is a versatile measure.

As expected the information content and path based measures generally struggled. This can be largely explained by the fact that the 628 *is–a* hierarchies for verbs are very shallow, the vast majority having from two to five levels. For the nouns there are nine rather deep hierarchies, and then these are joined together by a single artificial root. This creates a single, rich tree structure that has a maximum depth of 16 levels. As such, information content and path

27

Table 4
F–Measure (768 instances of 15 adjectives)

	Window Size					
Measure	2	3	5	11	21	51
random	.186	.176	.156	.174	.186	.179
edge	.045	.043	.047	.046	.047	.045
hso	.044	.078	.084	.098	.032	.031
lesk-o	.082	.133	.159	.185	.175	.183
lesk-e	.197	.245	.235	**.243**	**.232**	**.229**
vector	**.234**	**.251**	.240	**.243**	.224	.212

based are much more effective for nouns in WordNet.

However, jcn-sem fares reasonably well relative to the other measures. As was the case with nouns, it is the only information content measure that performs substantially differently when information content is computed from the British National Corpus versus SemCor. The lin and res measures do not demonstrate any particular difference in performance despite using radically different sources for information content computations.

8.3 Adjectives

Adjectives in WordNet are not arranged in a hierarchy, which prevents path based and information content measures from being applied. However, adjectives have glosses associated with their senses in WordNet, so gloss based measures are useful. We show these results in Table 4. By way of comparison, we have also included the edge measure as a representative of the path based measure, all of which fared quite poorly due to the structural limitations of WordNet.

For adjectives we observe that vector and lesk-e perform quite well, with vector attaining the highest F–measure of .251. These two measures fare substantially better than edge or hso, which is not surprising given how few relations there are to and from adjectives. This demonstrates the flexibility of gloss based measures, in that they do not require that relations be explicitly encoded in order to perform well.

Table 5
Overall Results (all 4,328 instances)

Measure	Window Size					
	2	3	5	11	21	51
random	.156	.147	.147	.167	.165	.158
edge	.128	.150	.159	.169	.166	.163
lesk-o	.096	.145	.174	.200	.200	.197
lesk-e	**.231**	**.278**	**.284**	**.295**	**.292**	**.277**
vector	.217	.232	.230	.241	.237	.222

8.4 Overall Results

Only the gloss based measures and hso are designed to measure the relatedness of nouns, verbs, and adjectives. We have previously observed that hso has struggled with all parts of speech, so we do not include those overall results here. We believe that this reflects more upon the somewhat impoverished relation structure in WordNet, more than it does upon the intrinsic merit of hso.

We show overall results for the gloss based measures and our baselines in Table 5. These results show that the extended gloss overlaps (lesk-e) is overall our most effective measure, attaining a maximum F–measure of .295.

We also note that for all parts of speech, the vector measure results in much higher F–measures than does the traditional gloss overlap measure of Lesk (lesk-o). This shows the value of measuring relatedness via second–order co–occurrences, rather than first order ones as lesk-o relies upon.

Since we are using the same data as was used in Senseval-2, it is possible to make direct comparisons with the results attained by lesk-e, vector, and those results. The best reported result for the unsupervised English lexical sample task in Senseval-2 was an overall F-measure of .402. While this is somewhat higher than our reported best of .295, it is worth nothing that the best reported Senseval-2 value reflects the combination of two different techniques in a single approach.

This system (UNED-LS-U [29]) incorporated information from a 277 million word corpus derived from Project Gutenberg to create a relevance matrix. A technique based on the relevance matrix was able to disambiguate 3,039 of the 4,328 instances, and attained precision of .373 and recall of .262, for an F–measure of .308. This is in fact comparable to our lesk-e value of .295.

However, the UNED-LS-U system was unable to disambiguate the remaining 1,289 instances and so the system had a back–off strategy such that the unattempted instances were assigned the first listed sense for the target word in WordNet. This resulted in precision of .467 and recall of .139, or an F–measure of .214. When these two approaches were combined, the overall results were quite good. It suggests that we should consider a combination scheme for our methods, which in fact we propose to do in future work.

The second place system (CL Research–DIMAP) relied on WordNet and achieved an F–Measure of .293 [30]. It attained F–measure of .354 on adjectives, .338 on nouns, and .225 on verbs. These results are quite similar to our own overall, although we note that our approach is somewhat better for nouns, and less so for verbs and adjectives. Again, this suggests the value of combining systems.

Thus, we believe the results of attained by maximizing semantic relatedness with lesk-e compare favorably with some of the best previously reported results for this data. We believe that our approach has the added merit of being intuitively appealing, and it allows us to easily leverage the considerable body of work that has gone into developing measures of semantic relatedness. The disambiguation algorithm is in fact quite simple, so we believe that the results here suggest that measures of semantic relatedness are powerful sources of information for disambiguation in general.

9   Future Work

This method of disambiguation depends quite crucially on the quality of the measures of semantic relatedness that are used. All of the measures could be tuned and refined in various ways, and we mention a few examples here to give a flavor of such work. Information content values and co–occurrence matrices for the gloss vector measure could be calculated from different corpora, and various smoothing methods could be explored. Rather than relying on frequency counts they could employ measures of association. For the gloss overlap measures, fuzzy string matching strategies could be explored. It would be useful to determine which relations contribute the most to the extended gloss overlap success, rather than simply using all of the immediately available relations.

In the end it appears that all of these measures offer some unique capability, and exploring the possibility of combining them into ensembles is perhaps the most promising area of future work. There is considerable evidence already in word sense disambiguation that the combination of different knowledge sources will bring improvements (e.g., [31], [32]) and we see no reason to believe that

these measures are any exception.

The disambiguation algorithm itself could be refined in various ways. The contribution of relatedness could be weighted according to the distance from the target word (or other criteria). The selection of words in the window of context could be refined such that only the most relevant words are selected.

10   Conclusions

In this article we have introduced a method of word sense disambiguation that selects the sense of a target word that has the maximum relatedness with the content words found in a large window of surrounding context. We show that this algorithm can be used with any measure that computes a relatedness score between two concepts, and found that in general the performance of this algorithm improves as the window of context increases. We observed that the extended gloss overlap measure (lesk-e) is overall the most effective, and the the gloss vector (vector) measure fared particularly well with adjectives, which are essentially impossible for path based and information content measures. When we compare vector with lesk-o, we can quickly conclude that there are considerable advantages to using second order co–occurrences to carry out gloss matching versus more traditional first order overlaps. We also observed that the Jiang–Conrath measure (jcn) performs quite well for nouns and verbs, but only when its information content is computed from the sense–tagged corpus SemCor.

In general we believe that the results of this work show that measures of semantic relatedness are a useful and important knowledge source for word sense disambiguation.

11   Resources

The disambiguation algorithm and all the measures of relatedness described in this paper are available as freely distributed Perl programs. For these experiments, we used version 0.1 of the SenseRelate package, and version 0.05 of the WordNet::Similarity package to compute all the measures of relatedness reported. These can be found at:

- http://www.d.umn.edu/~tpederse/senserelate.html
- http://search.cpan.org/dist/WordNet-Similarity/

References

[1] S. Banerjee, 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, 2002, pp. 136–145.

[2] M. Lesk, Automatic sense disambiguation using machine readable dictionaries: how to tell a pine code from an ice cream cone, in: Proceedings of the 5th annual international conference on Systems documentation, ACM Press, 1986, pp. 24–26.

[3] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, 1995, pp. 448–453.

[4] S. Patwardhan, S. Banerjee, 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, Mexico City, 2003, pp. 241–257.

[5] Z. Wu, M. Palmer, Verb semantics and lexical selection, in: 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, 1994, pp. 133–138.

[6] C. Leacock, M. Chodorow, Combining local context and WordNet similarity for word sense identification, in: C. Fellbaum (Ed.), WordNet: An electronic lexical database, MIT Press, 1998, pp. 265–283.

[7] G. Hirst, D. St. Onge, Lexical chains as representations of context for the detection and correction of malapropisms, in: C. Fellbaum (Ed.), WordNet: An electronic lexical database, MIT Press, 1998, pp. 305–332.

[8] J. Jiang, D. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, in: Proceedings on International Conference on Research in Computational Linguistics, Taiwan, 1997, pp. 19–33.

[9] D. Lin, Using syntactic dependency as a local context to resolve word sense ambiguity, in: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, Madrid, 1997, pp. 64–71.

[10] S. Banerjee, T. Pedersen, Extended gloss overlaps as a measure of semantic relatedness, in: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, 2003, pp. 805–810.

[11] S. Patwardhan, Incorporating dictionary and corpus information into a context vector measure of semantic relatedness, Master's thesis, University of Minnesota, Duluth (August 2003).

[12] K. Sparck Jones, Synonymy and Semantic Classification, Edinburgh University Press, Edinburgh, 1986.

[13] M. Quillian, Semantic memory, in: M. Minsky (Ed.), Semantic Information Processing, The MIT Press, Cambridge, MA, 1968, pp. 227–270.

[14] Y. Wilks, D. Fass, C. Guo, J. McDonald, T. Plate, B. Slator, Providing machine tractable dictionary tools, Machine Translation 5 (1990) 99–154.

[15] J. Cowie, J. Guthrie, L. Guthrie, Lexical disambiguation using simulated annealing, in: Proceedings of the 14th International Conference on Computational Linguistics, Nantes, France, 1992, pp. 359–365.

[16] J. Veronis, N. Ide, Word sense disambiguation with very large neural networks extracted from machine readable dictionaries, in: Proceedings of the 13th International Conference on Computational Linguistics, Helsinki, 1990, pp. 389–394.

[17] H. Kozima, T. Furugori, Similarity between words computed by spreading activiation on an english dictionary, in: Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics, Utrecht, 1993, pp. 232–239.

[18] Y. Niwa, Y. Nitta, Co-occurrence vectors from corpora versus distance vectors from dictionaries, in: Proceedings of the Fifteenth International Conference on Computational Linguistics, Kyoto, Japan, 1994, pp. 304–309.

[19] 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, 1993, pp. 67–74.

[20] E. Agirre, G. Rigau, Word sense disambiguation using conceptual density, in: Proceedings of the 16th International Conference on Computational Linguistics, Copenhagen, 1996, pp. 16–22.

[21] C. Fellbaum (Ed.), WordNet: An electronic lexical database, MIT Press, 1998.

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

[23] P. Resnik, Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language, Journal of Artificial Intelligence Research 11 (1998) 95–130.

[24] A. Budanitsky, 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, 2001, pp. 29–34.

[25] P. Resnik, WordNet and class–based probabilities, in: C. Fellbaum (Ed.), WordNet: An electronic lexical database, MIT Press, 1998, pp. 239–263.

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

[27] G. Zipf, The Psycho-Biology of Language, Houghton Mifflin, Boston, MA, 1935.

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

[29] D. Fernandez-Amoros, J. Gonzalo, F. Verdejo, The UNED systems at senseval-2, in: Proceedings of the Senseval-2 Workshop, Toulouse, 2001, pp. 75–78.

[30] K. Litkowski, Use of machine readable dictionaries for word–sense disambiguation in senseval-2, in: Proceedings of the Senseval-2 Workshop, Toulouse, 2001, pp. 107–110.

[31] Y. Wilks, M. Stevenson, Word sense disambiguation using optimised combinations of knowledge sources, in: Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics, 1998, pp. 1398–1402.

[32] M. Stevenson, Y. Wilks, The interaction of knowledge sources in word sense disambiguation, Compuational Linguistics 27 (3) (2001) 321–349.