General Word Sense Disambiguation Method Based on a Full Sentential Context

Jiri Stetina Sadao Kurohashi Makoto Nagao
Graduate School of Infomatics, Kyoto University
Yoshida-honmachi, Sakyo, Kyoto, 606-8501, Japan
{stetina,kuro,nagao}@kuee.kyoto-u.ac.jp

Abstract

This paper presents a new general supervised word sense disambiguation method based on a relatively small syntactically parsed and semantically tagged training corpus. The method exploits a full sentential context and all the explicit semantic relations in a sentence to identify the senses of all of that sentence's content words. In spite of a very small training corpus, we report an overall accuracy of 80.3% (85.7, 63.9, 83.6 and 86.5%, for nouns, verbs, adjectives and adverbs, respectively), which exceeds the accuracy of a statistical sense-frequency based semantic tagging, the only really applicable general disambiguating technique.

1 Introduction

Identification of the right sense of a word in a sentence is crucial to any successful Natural Language Processing system. The same word can have different meanings in different contexts. The task of Word Sense Disambiguation is to determine the correct sense of a word in a given context.

In most cases the correct word sense can be identified using only the words co-occurring in the same sentence. However, very often we also need to use the context of words that appear outside the given sentence. For this reason we distinguish two types of contexts: the sentential context and the discourse context. The sentential context is given by the words which co-occur with the word in a sentence and by their relations to this word, while the discourse context is given by the words outside the sentence and their relations to the word. The problem that arises here is that most of the co-occurring words are also polysemous, and unless disambiguated they cannot fully contribute to the process of disambiguation. The senses of these words, however, also depend on the sense of the disambiguated word and therefore there is a reciprocal dependency which we will try to resolve by the algorithm described in this paper.

Table 1: Percentage of nouns, verbs, adjectives and adverbs and average number of senses

Category	Number	%	Average # of senses
NOUNS	48,534	45.5	5.4
VERBS	26,674	25.0	10.5
ADJECTIVES	19,743	18.5	5.5
ADVERBS	11,804	11.0	3.7
TOTAL	**106,755**	**100.0**	**5.8**

2 The Task Specification

For our work, we used the word sense definitions as given in WordNet (Miller, 1990), which is comparable to a good printed dictionary in its coverage and distinction of senses. Since WordNet only provides definitions for content words (nouns, verbs, adjectives and adverbs), we are only concerned with identifying the correct senses of the content words.

Both for the training and for the testing of our algorithm, we used the syntactically analysed sentences of the Brown Corpus (Marcus, 1993), which have been manually semantically tagged (Miller et al., 1993) into semantic concordance files (SemCor). These files combine 103 passages of the Brown Corpus with the WordNet lexical database in such a way that every content word in the text carries both a syntactic tag and a semantic tag pointing to the appropriate sense of that word in WordNet. Passages in the Brown Corpus are approximately 2,000 words long, and each contains approximately 1,000 content words.

The percentages of the nouns, verbs, adjectives and adverbs in the semantically tagged corpus, together with their average number of WordNet senses, are given in Table 1. Although most of the words in a dictionary are monosemous, it is the polysemous words that occur most frequently in speech and text. For example, over 80% of words in WordNet are monosemous, but almost 78% of the content words in the tested corpus had more than one sense, as shown in Table 2.

Table 2: Percentage of polysemous word in the corpus

Category	Number	Polysemous	%
NOUNS	48,534	38,279	78.9
VERBS	26,674	24,845	93.1
ADJECTIVES	19,743	13,315	67.4
ADVERBS	11,804	6,715	56.9
TOTAL	**106,755**	**83,154**	**77.9**

Assigning the most frequent sense (as defined by WordNet) to every content word in the used corpus would result in an accuracy of 75.2 %. Our aim is to create a word sense disambiguation system for identifying the correct senses of all content words in a given sentence, with an accuracy higher than would be achieved solely by a use of the most frequent sense.

3 General Word Sense Disambiguation

The aim of the system described here is to take any syntactically analysed sentence on the input and assign each of its content words a pointer to an appropriate sense in WordNet. Because the words in a sentence are bound by their syntactic relations, all the word's senses are determined by their most probable combination in all the syntactic relations derived from the parse structure of the given sentence. It is assumed here that each phrase has one central constituent (head), and all other constituents in the phrase modify the head (modifiers). It is also assumed that there is no relation between the modifiers. The relations are explicitly present in the parse tree, where head words propagate up through the tree, each parent receiving its head word from its head-child. Every syntactic relation can be also viewed as a semantic relationship between the concepts represented by the participating words. Consider, for example, the sentence (1) whose syntactic structure is given in Figure 1.

(1) The Fulton County Grand Jury said Friday an investigation of Atlanta's recent primary election produced no evidence that any irregularities took place.

Each word in the above sentence is bound by a number of syntactic relations which determine the correct sense of the word. For example, the sense of the verb produced is constrained by the subject-verb relation with the noun investigation, by the verb-object relation with the noun evidence and by the subordinate clause relation with the verb said. Similarly, the verb said is constrained by its relations with the words Jury, Friday and produced; the sense of the noun investigation is constrained by the relation with the head of its prepositional phrase - election, and by the subject-verb relation with the verb produced, and so on.

The key to extraction of the relations is that any phrase can be substituted by the corresponding tree head-word (links marked bold in Figure 1). To determine the tree head-word we used a set of rules similar to that described by (Magerman, 1995)(Jelinek et al., 1994) and also used by (Collins, 1996), which we modified in the following way:

- The head of a prepositional phrase (PP→ IN NP) was substituted by a function the name of which corresponds to the preposition, and its sole argument corresponds to the head of the noun phrase NP.
- The head of a subordinate clause was changed to a function named after the head of the first element in the subordinate clause (usually 'that' or a 'NULL' element) and its sole argument corresponds to the head of its second element (usually head of a sentence).

Because we assumed that the relations within the same phrase are independent, all the relations are between the modifier constituents and the head of a phrase only. This is not necessarily true in some situations, but for the sake of simplicity we took the liberty to assume so. A complete list of applicable relations for sentence (1) is given in (2).

(2) NP(NNP(County),NNP(Jury))
NP(NNP(Grand),NNP(Jury))
NP(NP(Atlanta),NP(election))
NP(JJ(recent),NP(election))
NP(JJ(primary),NN(election))
NP(NN(investigation),PP(of(election)))
S(NP(irregularities),VP(took))
VP(VBD(took),NP(place))
NP(NN(evidence),SBAR(that(took))
S(NP(investigation),VP(produced))
VP(VBD(produced),NP(evidence))
VP(VBD(said),NP(Friday))
VP(VBD(said),SBAR(0(produced)))
S(NP(Jury),VP(said))

Each of the extracted syntactic relations has a certain probability for each combination of the senses of its arguments. This probability is derived from the probability of the semantic relation of each combination of the sense candidates of the related content words. Therefore, the approach described here consists of two phases: 1. learning the semantic relations, and 2. disambiguation through the probability evaluation of relations.

4 Learning

At first, every content word in every sentence in the training set was tagged by an appropriate pointer to a sense in WordNet.

Secondly, using the parse trees of all the corpus sentences, all the syntactic relations present in the

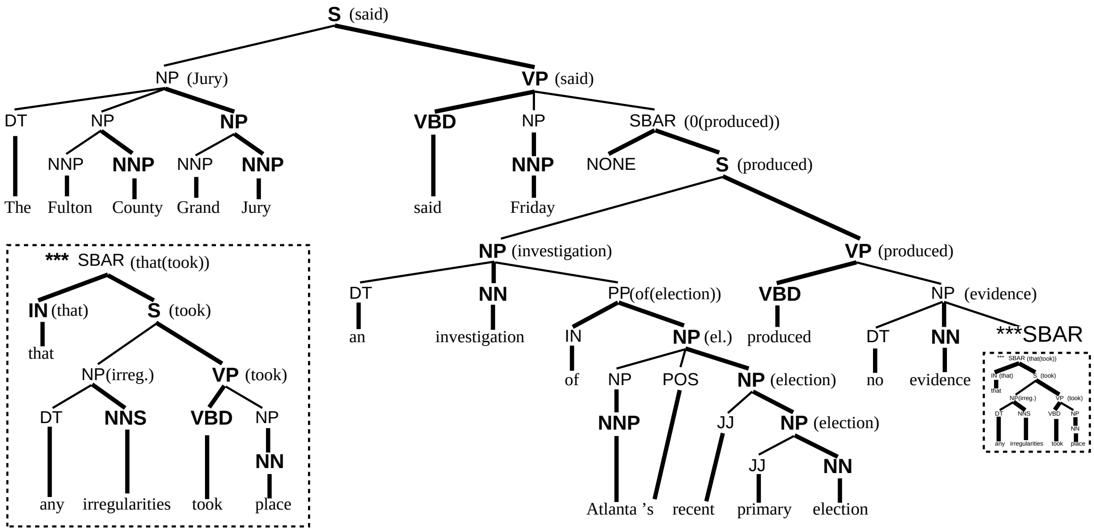

Figure 1: Example parse tree

training corpus were extracted and converted into the following form:

(4) rel(PNT, MNT, HNT, MS, HS, RP).

where PNT is the phrase parent non-terminal, MNT the modifier non-terminal, HNT the head non-terminal, MS the semantic content (see below) of the modifier constituent, HS the semantic content of the head constituent and RP the relative position of the modifier and the head (RP=1 indicates that the modifier precedes the head, while for RP=2 the head precedes the modifier). Relations involving non-content modifiers were ignored. Synsets of the words not present in WordNet were substituted by the words themselves.

The semantic content was either a WordNet sense identificator (synset) or, in the case of prepositional and subordinate phrases, a function of the preposition (or a null element) and the sense identificator of the second phrase constituent.

5 Disambiguation Algorithm

As mentioned above, we assumed that all the content words in a sentence are bound by a number of syntactic relations. Every content word can have several meanings, but each of these meanings has a different probability, which is given by the set of semantic relations in which the word participates. Because every relation has two arguments (head and its modifier), the probability of each sense also depends on the probability of the sense of the other participant in the relation. The task is to select such a combination of senses for all the content words, that the overall relational probability is maximal. If, for any given sentence, we had extracted **N** syntactic relations **Ri**, the overall relational probability for the combination of senses **X** would be:

$$(5)\, ORP(X) = \prod_{i=1}^{N} p(R_i|X)$$

where $\mathbf{p}(\mathbf{R_i}|\mathbf{X})$ is the probability of the i-th relation given the combination of senses **X**. If we consider, that an average word sense ambiguity in the used corpus is 5.8 senses, a sentence with 10 content words would have 5.8^{10} possible sense combinations, leading to a combinatorial explosion of over 43,080,420 overall probability combinations, which is not feasible. Also, with a very small training corpus, it is not possible to estimate the sense probabilities very accurately. Therefore, we have opted for a hierarchical disambiguation approach based on similarity measures between the tested and the training relations, which we will describe in Section 5.2. At first, however, we will describe the part of the probabilistic model which assigns probability estimates to the individual sense combinations based on the semantic relations acquired in the learning phase.

5.1 Relational Probability Estimate

Consider, for example, the syntactic relation between a head noun and its adjectival modifier derived from NP→ JJ NN. Let us assume that the number of senses in WordNet is **k** for the adjective and **l** for the noun. The number of possible sense combinations is therefore $m = k * l$. The probability estimate of a sense combination (i,j) in the relation **R**, where **i** is the sense of the modifier (adjective in this example) and **j** is the sense of the head (noun in this example), is calculated as follows:

$$(6)\, pR(i,j) = \frac{fR(i,j)}{\sum_{o=1}^{k}\sum_{p=1}^{l} fR(o,p)}$$

where **fR(i,j)** is a score of co-occurrences of a modifier sense **x** with a head word sense **y**, among the same semantic relations **R** extracted during the learning phase. Please note, that because **fR(i,j)** is not a count but rather a score of co-occurrences (defined below), **pR(i,j)** is not a real probability but rather its approximation. Because the occurrence count is replaced by a similarity score, the sparse data problem of a small training corpus is substantially reduced. The score of co-occurrences is defined as a sum of hits of similar pairs, where a hit is a multiplication of the similarity measures, sim(i,x) and sim(j,y), between both participants, i.e.:

$$(7)\, fR(i,j) = \sum_{q=1}^{r} sim(i,x) \cdot sim(j,y)$$

where $\mathbf{x}, \mathbf{y} \in \mathbf{R}$; **r** is the number of relations of the same type (for the above example R=rel(NP,ADJ,NOUN,x,y,1)) found in the training corpus. To emphasise the sense-restricting contribution of each example found, every pair (x,y) is restricted to contributing to only one sense combination (i,j): every example pair (x,y) contributes only to such a combination for which $sim(i,x) * sim(j,y)$ is maximal.

fR(i,j) represents a sum of all hits in the training corpus for the sense combination (i,j). Because the similarity measure (see below) has a value between 0 and 1 and each hit is a multiplication of two similarities, its value is also between 0 and 1. The reason why we used a multiplication of similarities was to eliminate the contributions of examples in which one participant belonged to a completely different semantic class. For example, the training pair *new airport*, makes no contribution to the probability estimate of any sense combination of *new management*, because none of the two senses of the noun *management* (group or human activity) belongs to the same semantic class as *airport* (entity). On the other hand, *new airport* would contribute to the probability estimate of the sense combination of *modern building* because one sense of the adjective *modern* is synonymous to one sense of the adjective *new*, and one sense of the noun **building** belongs to the same conceptual class (entity) as the noun *airport*. The situation is analogous for all other relations. The reason why we used a count modified by the semantic distances, rather than a count of exact matches only, was to avoid situations where no match would be found due to the sparse data, a problem of many small training corpora.

Every semantic relation can be represented by a **relational matrix**, which is a matrix whose first coordinate represents the sense of the modifier, the second coordinate represents the sense of the head and the value at the coordinate position (i,j) is the estimate of the probability of the sense combination (i,j) computed by (6). An example of a relational matrix for an adjective-noun relation *modern building* based on two training examples (*new airport* and *classical music*) is given in Figure 3. Naturally, the more the examples, the more fields of the matrix get filled. The training examples have an accumulative effect on the matrix, because the sense probabilities in the matrix are calculated as a sum of 'similarity based frequency scores' of all examples (7) divided by the sum of all matrix entries, (6). The most likely sense combination scores the highest value in the matrix. Each semantic relation has its own matrix. The way all the relations are combined is described in Section 5.2.

5.1.1 Semantic Similarity

We base the definition of the semantic similarity between two concepts (concepts are defined by their WordNet synsets a,b) on their semantic distance, as follows:

$$(8)\, sim(a,b) = 1 - sd(a,b)^2,$$

The semantic distance **sd(a,b)** is squared in the above formula in order to give a bigger weight to closer matches.

The semantic distance is calculated as follows.

Semantic Distance for Nouns and Verbs

$$sd(a,b) = \frac{1}{2} \cdot \left(\frac{D1 - D}{D1} + \frac{D2 - D}{D2}\right)$$

where D1 is the depth of synset a, D2 is the depth of synset D2, and D is the depth of their nearest common ancestor in the WordNet hierarchy. If a and b have no common ancestor, sd(a,b) = 1.

If any of the participants in the semantic distance calculation is a function (derived from a prepositional phrase or subordinate clause), the distance is equal to the distance of the function arguments for the same functor, or equals 1 for different functors. For example, sd(of(sense1), of(sense2)) = sd(sense1, sense2), while sd(of(sense1), about(sense2)) = 1, no matter what sense1 and sense2 are.

Semantic Distance for Adjectives

sd(a,b) = 0 for the same adjectival synsets (incl.synonymy),
sd(a,b) = 0 for the synsets in antonymy relations, i.e. for ant(a,b),
sd(a,b) = 0.5 for the synsets in the same similarity cluster,
sd(a,b) = 0.5 if a belongs to the same similarity cluster as c and b is the antonymy of c (indirect antonymy),
sd(a,b) = 1 for all other synsets.

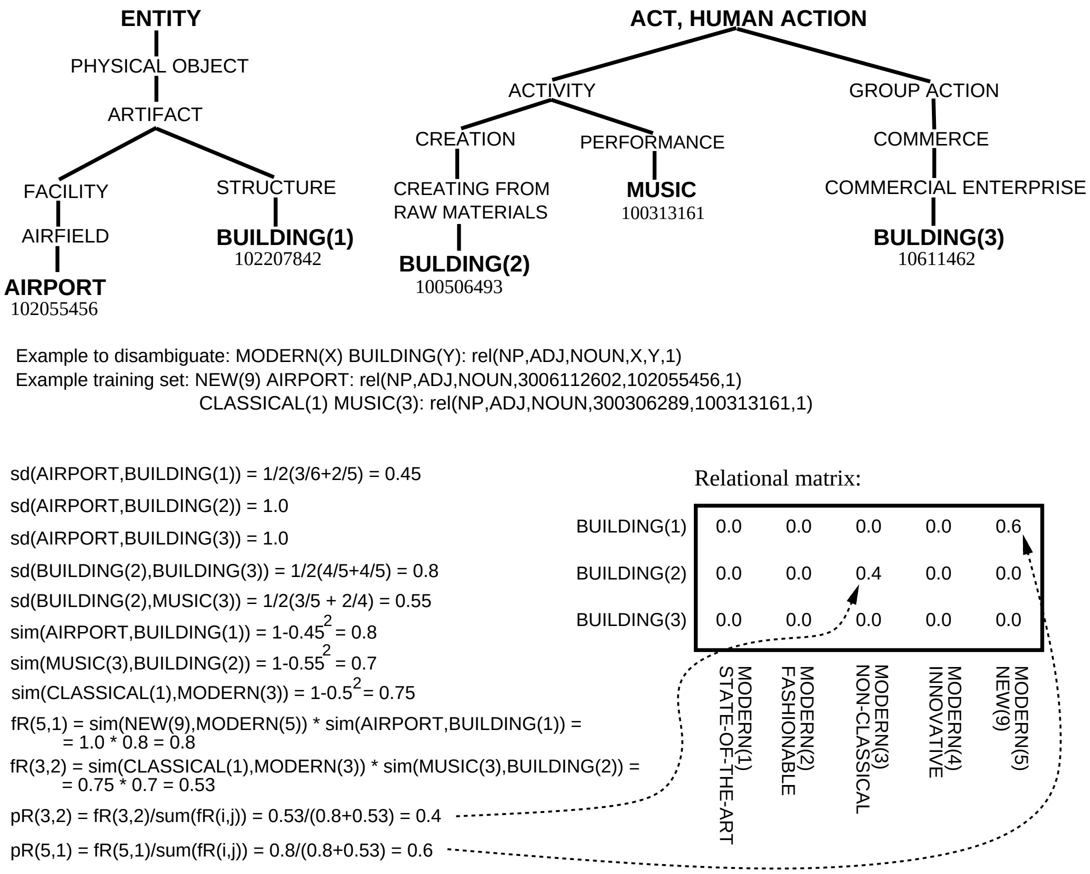

Figure 2: Relational matrix based on two training examples

Semantic Distance for Adverbs

sd(a,b) = 0 for the same synsets (incl.synonymy),
sd(a,b) = 0 for the synsets in antonymy relation ant(a,b),
sd(a,b) = 1 for all other synsets.

5.2 Hierarchical Disambiguation

This section describes the main part of the algorithm, i.e. the disambiguation process based on the overall probability estimate of sentential relations. As we have outlined above, for computational reasons, it is not feasible to evaluate overall probabilities for all the sense combinations. Instead, we take advantage of the hierarchical structure of each sentence and arrive at the optimum combination of its word senses, in a process which has two parts: 1. bottom-up propagation of the head word sense scores and 2. top-down disambiguation.

5.2.1 Bottom-up head word sense score propagation

In compliance with our assumption that all the semantic relations are only between a head word and its modifiers at any syntactic level, the modifiers do not participate in any relation with an element outside their parent phrase. As depicted in the example in Figure 1, it is only the head word concepts that propagate through the parse tree and that participate in semantic relations with concepts on other levels of the parse tree. The modifiers (which are heads themselves at lower tree levels), however, play an important role in constraining the head-word senses. The number of relations derived at each level of the tree depends on the number of concepts that modify the head. Each of these relations contributes to the score of each sense of the head word. We define the **sense score vector** of a word **w** as a vector of scores of each WordNet sense of the word **w**. The **initial sense score vector** of the word **w** is given by its contextually independent sense distribution in the whole training corpus. Because the training corpus is relatively small, and because it always excludes the tested file, an appropriate sense of the word w may not be present in it at all. Therefore, each sense **i** of the word **w** is always given a non-zero initial score $\mathbf{p_i(w)}$ (9a):

$$(9a)\, p_i(w) = \frac{count(w)_i + 1}{\sum_{j=1}^{n}(count(w)_j + 1)}$$

where $count(w)_i$ is the number of occurrences of the sense i of the word **w** in the entire training corpus, and **n** is the number of different WordNet senses of the word **w**.

The sense score vectors of head words propagate up the tree. At each level, they are modified by all the semantic relations with their modifiers which occur at that level. Also, the sense score vectors of head words are used to calculate the matrices of the sense score vectors of the modifiers. This is done as follows:

Let $\mathbf{H} = [\mathbf{h1}, \mathbf{h2}, ..., \mathbf{hk}]$ be the sense score vector of the head word **h**. Let $\mathbf{T} = [\mathbf{R1}, \mathbf{R2}, ...\mathbf{Rn}]$ be a set of relations between the head word **h** and its modifiers.

1. For each semantic relation $\mathbf{Ri} \in \mathbf{T}$ between the head word **h** and a modifier **mi** with sense score vector $\mathbf{Mi} = [\mathbf{oi1}, \mathbf{oi2}, ...\mathbf{oil}]$, do:
   1.1 Using (6), calculate the relational matrix **Ri(m,h)** of the relation **Ri**
   1.2 For each $\mathbf{oi} \in \mathbf{Mi}$ multiply all the elements of the **Ri(m,h)** for which **m=oi** by **oi**, yielding **Qi** - the **sense score matrix** of the modifier **mi**
2. The new sense score vector of the head word **h** is now **G=** $[\mathbf{g1}, \mathbf{g2}, ..., \mathbf{gk}]$, where

$$(10)\, g_j = \frac{L_j}{L} * h_j$$

$\mathbf{L_j/L}$ represents the score of the head word sense **j** based on the matrices **Q** calculated in the step 1., i.e.:

$$(11)\, Lj = \sum_{i=1}^{n} max(xi(j,u))$$

where $\mathbf{xi(j,u)} \in \mathbf{Qi}$ and $\mathbf{max(xi(j,u))}$ is the highest score in the line of the matrix **Qi** which corresponds to the head word sense **j**. **n** is the number of modifiers of the head word **h** at the current tree level, and

$$Lj = \sum_{j=1}^{k} Lj$$

where **k** is the number of senses of the head word **h**.

The reason why $\mathbf{g_j}$ (10) is calculated as a sum of the best scores (11), rather than by using the traditional maximum likelihood estimate (Berger et al., 1996)(Gale et al., 1993), is to minimise the effect of the sparse data problem. Imagine, for example, the phrase VP→ VB NP PP, where the head verb VB is in the object relation with the head of the noun phrase NP and also in the modifying relation with the head of the prepositional phrase PP. Let us also assume that the correct sense of the verb VB is **a**. Even if the verb-object relation provided a strong selectional support for the sense **a**, if there was no example in the training set for the second relation (between VB and PP) which would score a hit for the sense **a**, multiplying the scores of that sense derived from the first and from the second relation respectively, would gain a zero probability for this sense and thus prevent its correct assignment.

The newly created head word sense score vector **G** propagates upwards in the parse tree and the same process repeats at the next syntactic level. Note that at the higher level, depending on the head extraction rules described in section 3, the roles may be changed and the former head word may become a modifier of a new head (and participate in the above calculation as a modifier). The process repeats itself until the root of the tree is reached. The word sense score vector which has reached the root, represents a vector of scores of the senses of the main head word of the sentence (verb said in the example in Figure 1), which is based on the whole syntactic structure of that sentence. The sense with the highest score is selected and the sentence head disambiguated.

5.2.2 Top-down Disambiguation

Having ascertained the sense of the sentence head, the process of top-down disambiguation begins. The top-down disambiguation algorithm, which starts with the sentence head, can be described recursively as follows:

Let **l** be the sense of the head word **h** on the input. Let $\mathbf{M} = [\mathbf{m1}, \mathbf{m2}, ..., \mathbf{mx}]$ be the set of the modifiers of the head word **h**. For every modifier $\mathbf{mi} \in \mathbf{M}$, do:

1. In the sense score matrix **Qi** of the modifier **mi** (calculated in step 1.2 of the bottom-up phase) find all the elements **x(ki,l)**, where **l** is the sense of the head **h**
2. Assign the modifier **mi** such a sense **k=ki** for which the value **x(ki,l)** is maximum. In the case of a draw, choose the sense which is listed as more frequent in WordNet.
3. If the modifier **mi** has descendants in the parse tree, call the same algorithm again with **mi** being the head and **k** being its sense, else **end**.

The disambiguation of the modifiers (which become heads at lower levels of the parse tree), is based solely on those lines of their sense score matrices which correspond to the sense of the head they are in relation with. This is possible because of our assumption that the modifiers are related only to their head words, and that there is no relation among the modifiers themselves. To what extent this assump-

Table 3: Number of words with the same and different sense as its previous occurrence in the same discourse (shortened)

	Has predecessor with the same sense				Has predecessor with a different sense			
Distance	NOUNS	VERBS	ADJs	ADVs	NOUNS	VERBS	ADJs	ADVs
anywhere	15,373	6,923	5,523	3812	2,057	5,227	933	830
<10	9,474	3,697	2,733	1672	649	2,521	258	214
<5	6,892	2,426	1,834	1000	355	1,561	104	135
<4	5,964	2,065	1,566	841	290	1,269	104	82
<3	4,797	1,578	1,219	614	208	929	83	55
<2	3,039	986	733	348	103	555	42	27

tion holds in real life sentences, however, has yet to be investigated.

6 Discourse Context

(Yarowsky, 1995) pointed out that the sense of a target word is highly consistent within any given document (one sense per discourse). Because our algorithm does not consider the context given by the preceding sentences, we have conducted the following experiment to see to what extent the discourse context could improve the performance of the word-sense disambiguation:

Using the semantic concordance files (Miller et al., 1993), we have counted the occurrences of content words which previously appear in the same discourse file. The experiment indicated that the "one sense per discourse" hypothesis works fairly well for nouns, however, the evidence is much weaker for verbs, adverbs and adjectives. Table 3 shows the numbers of content words which appear previously in the same discourse with the same meaning (same synset), and those which appear previously with a different meaning. The experiment also confirmed our expectation that the ratio of words with the same sense to those with a different sense, depends on the distance of sentences in which the same words appear (distance 1 indicates that the same word appeared in the previous sentence, distance 2 that the same word was present 2 sentences before, etc.).

We have modified the disambiguation algorithm to make use of the information gained by the above experiment in the following way: All the disambiguated words and their senses are stored. The words of all the input sentences are first compared with the set of these stored word-sense pairs. If the same word is found in the set, the initial sense score assigned to it by (9a) is modified using Table 3, so that the sense, which has been previously assigned to the word, gets higher priority. The calculation of the initial sense score (9a) is thus replaced by (9b):

$$(9b)\, p_i(w) = \frac{count(w)_i + 1}{\sum_{j=1}^{n}(count(w)_j + 1)} * e(POS, SN)$$

Table 4: Result Accuracy [%]

CONTEXT	NOUNS	VERBS	ADJs	ADVs	TOTAL
First sense	77.8	61.7	81.9	84.5	75.2
Sentence	84.2	63.6	82.9	86.3	79.4
+Discourse	85.7	63.9	83.6	86.5	80.3

where e(POS,SN) is the probability that the word with syntactic category POS which already occurred SN sentences before, has the same sense as its previous occurrence. If, for example, the same noun has occurred in the previous sentence (SN=1) where it was assigned sense n, the probability of sense n of the same noun in the current sentence is multiplied by e(NOUN,1)=3,039/(3,039+103)=0.967, while all the probabilities of its remaining senses are multiplied by 1-0.967=0.033. If no match is found, i.e. the word has not previously occurred in the discourse, e(POS,SN) is set to 1 for all senses.

7 Evaluation

To evaluate the algorithm, we randomly selected 15 files (with a total of 18,413 content words tagged in SemCor) from the set of 103 files of the sense tagged section of the Brown Corpus. Each tested file was removed from the set and the remaining 102 files were used for learning (Section 4). Every sense assigned by the hierarchical disambiguation algorithm (Section 5) was compared with the sense from the corresponding semantic concordance file. Table 4 shows the achieved accuracy compared with the accuracy which would be achieved by a simple use of the most frequent sense.

As the above table shows, the accuracy of the word sense disambiguation achieved by our method was better than using the first sense for all lexical categories. In spite of a very small training corpus, the overall word sense accuracy exceeds 80%.

8 Related Work

To our knowledge, there is no current method which attempts to identify the senses of all words in whole sentences, so we cannot make a practical comparison.

Similarly to our work, (Resnik, 1995)(Agirre and Rigau, 1996) challenge the fine-grainedness of WordNet, but their work is limited to nouns only. (Agirre and Rigau, 1996) report coverage 86.2%, precision 71.2% and recall 61.4% for nouns in four randomly selected semantic concordance files. From among the methods based on semantic distance, (Resnik, 1993)(Sussna, 1993) use a similar semantic distance measure for two concepts in WordNet, but they also focus on selected group of nouns only. (Karov and Edelman, 1996) use an interesting iterative algorithm and attempt to solve the sparse data bottleneck by using a graded measure of contextual similarity. They achieve 90.5, 92.5, 94.8 and 92.3 percent accuracy in distinguishing between two senses of the noun drug, sentence, suit and player, respectively. (Yarowsky, 1995), whose training corpus for the noun drug was 9 times bigger than that of Karov and Edelman, reports 91.4% correct performance improved to impressive 93.9% when using the "one sense per discourse" constraint. These methods, however, focus on only two senses of a very limited number of nouns and therefore are not comparable with our approach.

9 Conclusion

This paper presents a new general approach to word sense disambiguation. Unlike most of the existing methods, it identifies the senses of all content words in a sentence based on an estimation of the overall probability of all semantic relations in that sentence. By using the semantic distance measure, our method reduces the sparse data problem since the training examples and their contexts do not have to match the disambiguated words exactly. All the semantic relations in a sentence are combined according to the syntactic structure of the sentence, which makes the method particularly suitable for integration with a statistical parser into a powerful Natural Language Processing system. The method is designed to work with any type of common text and is capable of distinguishing among many word senses. It has a very wide scope of applicability and is not limited to only one part-of-speech.

References

E. Agirre and G. Rigau. 1996. Word sense disambiguation using conceptual density. In *Proc. of COLLING*, pages 16–22.

A. Berger, V. Pietra, and S. Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 1(22):39–72.

M. Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proc. of the 34th Annual Meeting of the ACL*, pages 184–191.

W. Gale, K. Church, and D. Yarowsky. 1993. A method for disambiguating word senses in a large corpus. *Computers and Humanities*, (26):415–4397.

F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, A. Rathnaparkhi, and S. Roukos. 1994. Decision tree parsing using a hidden derivation model. In *Proc. of the ARPA Human Language Technology Workshop*, pages 272–277.

Y. Karov and S. Edelman. 1996. Learning similarity-based word sense disambiguation from sparse data. In *Proc. of the 3rd Workshop on Very Large Corpora*, pages 42–55.

D. Magerman. 1995. Statistical decision-tree models for parsing. In *Proc. of the 33rd Annual Meeting of ACL*, pages 276–283.

M. Marcus. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 2(19):313–330.

G. Miller, C. Leacock, and R. Tengi. 1993. A semantic concordance. In *Proc. of the ARPA Human Language Technology Workshop*, pages 303–308.

G Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

P. Resnik. 1993. Semantic classses and syntactic ambiguity. In *Proc. of the APRA Human Language Technology Workshop*, pages 278–283.

P. Resnik. 1995. Disambiguating noun groupings with respect to wordnet senses. In *Proc. of the 3rd Workshop on Very Large Corpora*.

M. Sussna. 1993. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proc. of Second International Conference on Information and Knowledge Management*, pages 67–74.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of the 32nd Annual Meeting of the ACL.*