Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary

Jiri Stetina **Makoto Nagao**

Department of Electronics and Communications
Kyoto University, Yoshida Honmachi, Kyoto 606, Japan
stetina@pine.kuee.kyoto-u.ac.jp, nagao@pine.kuee.kyoto-u.ac.jp

ABSTRACT

This paper deals with two important ambiguities of natural language: prepositional phrase attachment and word sense ambiguity. We propose a new supervised learning method for PP-attachment based on a semantically tagged corpus. Because any sufficiently big sense-tagged corpus does not exist, we also propose a new unsupervised context based word sense disambiguation algorithm which amends the training corpus for the PP attachment by word sense tags. We present the results of our approach and evaluate the achieved PP attachment accuracy in comparison with other methods.

1. INTRODUCTION

The problem with successful resolution of ambiguous prepositional phrase attachment is that we need to employ various types of knowledge. Consider, for example, the following sentence:

1. *Buy books for children.*

The PP *for children* can be either adjectival and attach to the object noun *books* or adverbial and attach to the verb *buy*, leaving us with the ambiguity of two possible syntactic structures:

adj) VP(VB=buy, NP(NNS=books, PP(IN=for,NP(NN=children))))
adv) VP(VB=buy, NP(NNS=books), PP(IN=for,NP(NN=children))).

It is obvious that without some contextual information we cannot disambiguate such a sentence correctly. Consider, however, the next sentence:

2. *Buy books for money.*

In this case, we can almost certainly state that the PP is adverbial, i.e. attached to the verb. This resolution is based on our life time experience in which we much more often encounter the activity which can be described as "*buying things for money*" than entities described as "*books for money*"[1].

At the moment, we do not have a computer database containing life time experiences, and therefore we have to find another way of how to decide the correct PP attachment. One of the solutions lies in the exploration of huge textual corpora, which can partially substitute world knowledge. Partially, because we do not know how wide a context, what type of general knowledge or how deep an inference has to be applied for a successful disambiguation.

If we limit the context around the prepositional phrase to include only the verb, its object and the PP itself, the human performance on PP attachment is approximately 88.2% accurate decisions

[1] This, of course, does not provide us with a hundred percent certainty.

[RRR94]. Because people are capable of utilising their world knowledge, the remaining inaccuracy must be attributed to the lack of a wider context[2]. Statistically, each preposition has a certain percentage of occurrences for each attachment, relying on which would provide us with approximately 72.7% of correct attachments [C&B95]. If we manage to partially substitute the world knowledge, the resulting accuracy would lie in the range between 72.7 and 88.2%. These are the boundaries we expect an automatic system to score within.

1.1. PP-ATTACHMENT

Altman and Steedman [A&S88] have shown that in many cases PP can be attached correctly only if the context of the current discourse is used. Using the discourse context is, however, extremely difficult because we do not have enough theoretical background to decide which bits of context are needed to correctly disambiguate and which are irrelevant.

There have been numerous attempts to substitute context by superficial knowledge extracted from a large corpus. Pioneering research on corpus-based statistical PP attachment ambiguity resolution has been done by Hindle and Rooth in [H&R93]. They extracted over 200,000 verb-noun-preposition triples with unknown attachment decisions. An iterative, unsupervised method was then used to decide between adjectival and adverbial attachment in which the decision was based on comparing the co-occurence probabilities of the given preposition with the verb and with the noun in each quadruple.

Another promising approach is the transformation-based rule derivation presented by Brill and Resnik in [B&R94], which is a simple learning algorithm that derives a set of transformation rules. These rules are then used for PP attachment and therefore, unlike the statistical methods, it is unnecessary to store huge frequency tables. Brill and Resnik had reported 81.8% success of this method on 500 randomly-selected sentences.

The current statistical state-of-the art method is the backed-off model proposed by Collins and Brooks in [C&B95] which performs with 84.5% accuracy on stand-alone quadruples. Most of the methods, however, suffer from a sparse data problem. All are based on matching the words from the analysed sentence against the words in the training set. The problem is that only exact matches are allowed. The back-off model showed an overall accuracy of 84.5%, but the accuracy of full quadruple matches was 92.6%! Due to the sparse data problem, however, the full quadruple matches were quite rare, and contributed to the result in only 4.8% of cases. The accuracy for a match on three words was also still relatively high (90.1%), while for doubles and singles it dropped substantially [C&B95].

This originated our assumption that if the number of matches on four and three words was raised, the overall accuracy would increase as well. Because Collins and Brooks' backing-off model is very profound, we could not find a way of improving its accuracy unless we increased the percentage of full quadruple and triple matches by employing the semantic distance measure instead of word-string matching. We feel that the sentence *Buy books for children* should be matched with *Buy magazines for children* due to the small conceptual distance between books and magazines. What is unknown, however, is the limit distance for two concepts to be matched. Many nouns in the WordNet hierarchy share the same root (entity) and there is a danger of over-generalisation. We will try to overcome this problem through the supervised learning algorithm described herein. Another problem is that most of the words are semantically ambiguous and unless disambiguated, it is difficult to establish distances between them. The PP attachment also depends on the selection of word senses and vice versa, as will be shown in the result section.

A number of other researchers have explored corpus-based approaches to PP attachment that make

[2] Human performance on the same data but with a full sentential context is 93.2% [RRR94].

use of word classes. For examples, Weischedel [W91] and Basili [B91] both describe the use of manually constructed, domain specific word classes together with corpus-based statistics in order to resolve PP attachment ambiguity. Because these papers describe results obtained on different corpora, however, it is difficult to make a performance comparison.

1.2 WORD SENSE AMBIGUITY

We will now discuss the issues connected with matching two different words based on their semantic distance. Employing the notion of semantic similarity, it is necessary to address a number of problems. At first, we have to specify the semantic hierarchy. Second, we need to determine how to calculate the distance between two different concepts in the hierarchy. Finally we must determine how to select a sense of a word based on a context in which it appears.

SEMANTIC HIERARCHY

The hierarchy we chose for semantic matching is the semantic network of WordNet [MI90], [MI93]. WordNet is a network of meanings connected by a variety of relations. WordNet presently contains approximately 95.000 different word forms organised into 70.100 word meanings, or sets of synonyms. It is divided into four categories (nouns, verbs, adjectives and adverbs), out of which we will be using only verbs and nouns. Nouns are organised as 11 topical hierarchies, where each root represents the most general concept for each topic. Verbs, which tend to be more polysemous and can change their meanings depending on the kind of the object they take, are formed into 15 groups and have altogether 337 possible roots. Verb hierarchies are more shallow than those of nouns, as nouns tend to be more easily organised by the *is-a* relation, while this is not always possible for verbs.

SEMANTIC DISTANCE

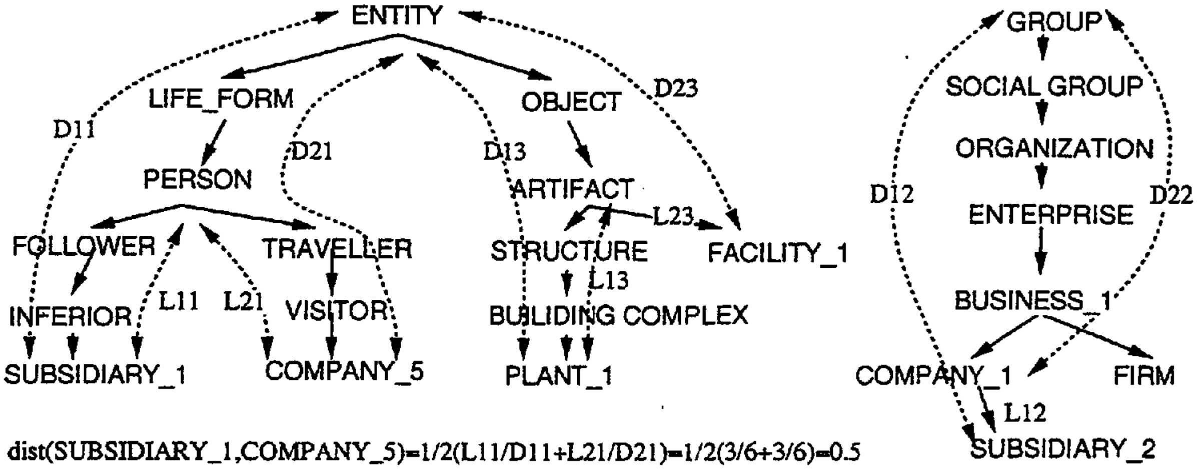
The traditional method of evaluating semantic distance between two meanings based merely on the length of the path between the nodes representing them, does not work well in WordNet, because the distance also depends on the depth at which the concepts appear in the hierarchy. For example, the root *entity* is directly followed by the concept of *life_form*, while a *sedan*, a type of a car, is in terms of path more distant from the concept of *express_train*, although they are both vehicles and therefore closer concepts. In the case of verbs, the situation is even more complex, because many verbs do not share the same hierarchy, and therefore there is no direct path between the concepts they represent. There have been numerous attempts to define a measure for semantic distance of WordNet contained concepts [RE95],[K&E96], [SU95], [SU96], etc.

For our purposes, we have based the semantic distance calculation on a combination of the path distance between two nodes and their depth. Having ascertained the nearest common ancestor in the hierarchy, we calculate the distance as an average of the distance of the two concepts to their nearest common ancestor divided by the depth in the WordNet Hierarchy:

$$D = \frac{1}{2}\left({}^{L_1}\!/_{D_1} + {}^{L_2}\!/_{D_2} \right)$$

where L_1, L_2 are the lengths of paths between the concepts and the nearest common ancestor, and D_1, D_2 are the depths of each concept in the hierarchy (the distance to the root). The more abstract the concepts are (the higher in hierarchy), the bigger the distance. The same concepts have a distance equal to 0; concepts with no common ancestor have a distance equal to 1. Because the verb hierarchy is rather shallow and wide, the distance between many verbal concepts is often equal to 1.

Figure 1: *Distance calculation example*



dist(SUBSIDIARY_1,COMPANY_5)=1/2(L11/D11+L21/D21)=1/2(3/6+3/6)=0.5

dist(SUBSIDIARY_1,COMPANY_1)=0 (no common ancestor)

dist(SUBSIDIARY_2,COMPANY_1)=1/2(L12/D12+L22/D22)=1/2(1/7+0/6)=0.071

dist(PLANT_1,FACILITY1)=1/2(L13/D13+L23/D23)=1/2(3/6+1/4)=0.375

1.2.3 SEMANTIC AMBIGUITY

In order to determine the position of a word in the semantic hierarchy, we have to determine the meaning of the word from the context in which it appears. For example, the noun *bank* can take any of the nine meanings defined in WordNet (financial institution, building, ridge, container, slope, etc.). It is not a trivial problem and has been approached by many researchers [GCY92], [YA93], [B&W94], [RE95], [YA95], [K&E96], [LI96], etc. We believe that the word sense disambiguation can be accompanied by PP attachment resolution, and that they complement each other. At the same time we would like to note, that PP attachment and sense disambiguation are heavily contextually dependent problems. Therefore, we know in advance that without incorporation of wide context, the full disambiguation will be never reached.

2. WORD SENSE DISAMBIGUATION

The supervised learning algorithm which we have devised for the PP attachment resolution, and which is discussed in Chapter 3, is based on the induction of a decision tree from a large set of training examples which contain verb-noun-preposition-noun quadruples with disambiguated senses. Unfortunately, at the time of writing this work, a sufficiently big corpus which was both syntactically analysed and semantically tagged did not exist. Therefore, we used the syntactically analysed corpus [MA93] and assigned the word senses ourselves. Manual assignment, however, in the case of a huge corpus would be beyond our capacity and therefore we devised an automatic method for an approximate word sense disambiguation based on the following notions:

Determining the correct sense of an ambiguous word is highly dependent on the context in which the word occurs. Even without any sentential context, the human brain is capable of disambiguating word senses based on circumstances or experience[3]. In natural language

[3] Also a kind of context.

processing, however, we rely mostly on the sentential contexts, i.e. on the surrounding concepts and relations between them. These two problems arise: **1.** The surrounding concepts are very often expressed by ambiguous words and a correct sense for these words also has to be determined. **2.** What relations and how deep an inference is needed for correct disambiguation is unknown.

We based our word-sense disambiguating mechanism on the premise that two ambiguous words usually tend to stand for their most similar sense if they appear in the same context. In this chapter we present a similarity-based disambiguation method aimed at disambiguating sentences for subsequent PP-attachment resolution. Similar contextual situations (these include information on the PP-attachment) are found in the training corpora and are used for the sense disambiguation. If, for example, the verb *buy* (4 senses) appears in the sentence:

The investor bought the company for 5 million dollars

and somewhere else in the training corpus there is a sentence[4]:

The investor purchased the company for 5 million dollars,

we can take advantage of this similarity and disambiguate the verb "buy" to its sense that is nearest to the sense of the verb *purchase*, which is not ambiguous.

The situation, however, might not be as simplistic as that, because such obvious matches are extremely rare even in a huge corpus. The first problem is that the sample verb in the training corpus may be also ambiguous. Which sense do we therefore choose? The second problem is that there may, in fact, be no exact match in the training corpus for the context surrounding words and their relations. To overcome both of these problems we have applied the concept of semantic distance discussed above. Every possible sense of all the related context words is evaluated and the best match chosen[5].

The proposed **unsupervised similarity-based iterative algorithm** for the word sense disambiguation of the training corpus looks as follows:

```
1. From the training corpus, extract all the sentences which contain a prepositional
phrase with a verb-object-preposition-description quadruple. Mark each quadruple with
the corresponding PP attachment (explicitly present in the parsed corpus).
2. Set the Similarity Distance Threshold SDT = 0
3. Repeat
    * for each quadruple Q in the training set:
        * for each ambiguous word in the quadruple:
          * among the remaining quadruples find a set S of similar quadruples
            (those with quadruple distance < SDT)
          * for each non-empty set S:
                    * choose the nearest similar quadruple from the set S
                    * disambiguate the ambiguous word to the nearest sense of the
                      corresponding word of the chosen nearest quadruple
    * increase the Similarity Distance Threshold SDT = SDT + 0.1
  Until all the quadruples are disambiguated or SDT = 3.
```

[4] Both sentences have adverbial PP attachment.

[5] Because our primary goal is PP attachment disambiguation, the related context words are those appearing in the verb-noun-preposition-noun quadruple.

The above algorithm can be described as iterative clustering, because at first, the nearest quadruples are matched and disambiguated. Then, the similarity distance threshold is raised, and the process repeats itself in the next iteration. If a word is not successfully disambiguated, it is assigned its first, i.e. the most frequent sense. The reason for starting with the best matches is that these tend to provide better disambiguations. Consider, for example, the following set of quadruples:

Q1. shut plant for week
Q2. buy company for million
Q3. acquire business for million
Q4. purchase company for million
Q5. shut facility for inspection
Q6. acquire subsidiary for million

At first, the algorithm tries to disambiguate quadruple Q1. Starting with the verb, the algorithm searches for other quadruples which have the quadruple distance (see below) smaller than the current similarity distance threshold. For SDT=0 this means only for quadruples with all the words with semantic distance = 0, i.e. synonyms. There are no matches found for Q1 and the algorithm moves to Q2, finding quadruple Q4 as the only one matching such criteria. The verb *buy* in Q2 is disambiguated to the sense which is nearest to the sense of *purchase* in Q4, i.e. min(dist(*buy,purchase*))=dist(BUY-1,PURCHASE-1)=0.0. The noun *company* cannot be disambiguated, because the matched nearest quadruple Q4 contains the same noun and such a disambiguation is not allowed; the description *million* is monosemous. Same process is called for all the remaining quadruples but further disambigaution with SDT=0 is not possible (the verb *purchase* in Q4 has only one sense in WordNet and therefore there is no need for disambiguation; the noun *company* cannot be disambiguated against the same word). The iteration threshold is increased by 0.1 and the algorithm starts again with the first quadruple. No match is found for Q1 for any word and we have to move to quadruple Q2. Its verb is already disambiguated, therefore the algorithm looks for all the quadruples which have the quadruple distance for nouns below the SDT of 0.1 and which contain similar nouns (see definition of similar below). The quadruple Q3 satisfies this criteria. Distances of all the combinations of senses of the noun *company* and *business* are calculated and the nearest match chosen to disambiguate the noun *company* in Q2:

min(dist(*company,business*))=dist(COMPANY-1, BUSINESS-1)=0.083

The algorithm then proceeds to the next quadruple, i.e. Q3. There are two quadruples which satisfy the similarity threshold for verbs: Q2 and Q4 (Q6 is not considered, because its verb is identical and therefore not similar). The verb *buy* in Q2 is already disambiguated and the distance to both Q2 and Q4 is the same, i.e.:

$$dqv(Q3,Q2)=dqv(Q3,Q4)=(0.25^2+0.083+0)/3=0.0485$$

where the minimum semantic distance between the nearest senses of the verb *acquire* and *buy* is:

min(dist(*acquire,buy*)=dist(ACQUIRE-1,BUY-1)=0.25

The verb *acquire* is disambiguated to the sense nearest to the sense of the verb *buy* and the algorithm proceeds to the noun *business* in Q3. The same two quadruples fall below the SDT for nouns, as

$$dqn(Q3,Q2)=dqv(Q3,Q4)=(0.25+0.007+0)/3=0.0857$$

and the noun *business* of Q3 is disambiguated to its sense nearest to the disambiguated sense of *company* in Q2. The verb in Q4 is monosemous, therefore the algorithm finds a set of similar quadruples for nouns (Q2 qualifies in spite if having the same noun (*company*), because it has already been disambiguated in the previous steps): Q2, Q3 and Q6. The nearest quadruple in this set is Q2 (dqn(Q4,Q2)=0) and the noun *company* in Q4 is disambiguated to the sense of the noun in Q2. The quadruple Q5 has no similar quadruples for the current SDT and therefore the next quadruple is Q6. Similarly to the above disambiguations, both its verb and noun are disambiguated. There is no further match for any quadruple and therefore SDT is increased to 0.2 and the algorithm starts with Q1 again (the quadruples Q2, Q3, Q4 and Q6 are already fully disambiguated). No matches are found for SDT=0.2 for neither Q1 or Q5. The algorithm iterates until SDT=0.6 which enables the disambiguation of the noun *plant* in Q1 to its sense nearest to the noun *facility* in Q5:

$$dqn(Q1,Q5)=(0+0.375^2+1/)2=0.57$$

as min(dist(*plant,facility*)=dist(PLANT-1,FACILITY-1)=0.375. Similarly, the noun *facility* in Q6 is disambiguated, whereas the descriptions in both Q1 and Q5 cannot be successfully disambiguated because only a very small set of quadruples was used in this example. In this case, both the description *week* and *inspection* would be assigned their most frequent senses, i.e. the first senses of WordNet. In case of a bigger training set, most of the quadruples get disambiguated, however, with increasing SDT the disambiguation quality decreases. The above example shows the importance of iteration, because starting with lower SDT guarantees better results. If, for example, there was no iteration cycle and the algorithm tried to disambiguate the quadruples in the order in which they appear, the quadruple Q1 would be matched with Q6 and all its words would be disambiguated to inappropriate senses. Such a wrong disambiguation would further force wrong disambiguations in other quadruples and the overall result would be substantially less accurate. Another advantage of this disambiguation mechanism is that the proper nouns, which usually refer to people or companies, can be also disambiguated. For example, an unknown name *ARBY* in quadruple:

acquire ARBY for million

is matched with disambiguated noun in Q6 and also disambiguated to the COMPANY-1 sense, rather than to,PERSON (note, that even if Q6 was not disambiguated, the COMPANY-1 sense of *subsidiary* is semantically closer to the sense company of ARBY and therefore, although possible, the disambiguation of ARBY to the first sense of *subsidiary* (PERSON) would be dismissed).

Similarity Distance Threshold defines the limit matching distance between two quadruples. The matching distance between two quadruples $\mathbf{Q_1=v_1\text{-}n_1\text{-}p\text{-}d_1}$ and $\mathbf{Q_2=v_2\text{-}n_2\text{-}p\text{-}d_2}$ is defined as follows (v=verb, n=noun, p=preposition, d=description noun):

$$\mathbf{D_{qv}(Q_1,Q_2)= (D(v_1,v_2)^2)+D(n_1,n_2)+D(d_1,d_2))/P}, \text{ when disambiguating verb}$$
$$\mathbf{D_{qn}(Q_1,Q_2= (D(v_1,v_2)+D(n_1,n_2)^2+D(d_1,d_2))/P}, \text{ when disambiguating noun}$$
$$\mathbf{D_{qd}(Q_1,Q_2)= (D(v_1,v_2)+D(n_1,n_2)+D(d_1,d_2)^2)/P}, \text{ when disambiguating description}$$

where **P** is the number of pairs of words in the quadruples which have a common semantic ancestor, i.e. **P** = 1, 2 or 3 (if there is no such a pair, $Dq = \infty$) and its purpose is to give higher priority to matches on more words. The distance of the currently disambiguated word is squared in order to have a bigger weight in the distance $\mathbf{D_q}$ (the currently disambiguated word must be different from the corresponding word in the matched quadruple[6] unless it has been previously disambiguated). The distance between two words $\mathbf{D(w_1,w_2)}$ is defined as the minimum semantic distance between all the possible senses of the words $\mathbf{w_1}$ and $\mathbf{w_2}$. Two quadruples are **similar**, if their distance is less or equal to the current Similarity Distance Threshold, and if the currently disambiguated word is similar to the corresponding word in the matched quadruple. Two words are **similar** if their semantic distance is less than 1.0 and if either their character strings are different or if one of the words has been previously disambiguated.

[6] The same words have the same sets of senses and therefore would not allow for disambiguation.

3. PP-ATTACHMENT

For the attachment of the prepositional phrases in unseen sentences, we have modified Quinlan's ID3 algorithm [Q86], [BR91] which belongs to the the family of inductive learning algorithms. Using a huge training set of classified examples, it uncovers the importance of the individual words (attributes) and creates a decision tree that is later used for classification of unseen examples[7]. The algorithm uses the concepts of the WordNet hierarchy as attribute values and creates the decision tree in the following way:

3.1 DECISION TREE INDUCTION

Let **T** be a training set of classified quadruples.

1. If all the examples in **T** are of the same PP attachment type (or satisfy the homogeneity termination condition, see below) then the result is a leaf labelled with this type,

> **else**
>
> **2.** Select the most informative attribute A among verb, noun and description among the attributes not selected so far (the attributes can be selected repeatedly after all of them were already used in the current subtree)
>
> **3.** For each possible value A_W of the selected attribute A construct recursively a subtree S_W calling the same algorithm on a set of quadruples for which A belongs to the same WordNet class as A_W.
>
> **4.** Return a tree whose root is A and whose subtrees are S_W and links between A and S_W are labelled A_W.

Let us briefly explain each step of the algorithm.

1. If the examples belong to the same class (set T is homogenous), the tree expansion terminates. However, such situation is very unlikely due to the non-perfect training data. Therefore, we relaxed the complete homogeneity condition by terminating the expansion when more than 77% of the examples in the set belonged to the same class (the value of 77% was set experimentally as it provided the best classification results). If the set T is still heterogeneous and there are no more attribute values to divide with, the tree is terminated and the leaf is marked by the majority class of the node.

2. We consider the most informative attribute to be the one which splits the set T into the most homogenous subsets, i.e. subsets with either a high percentage of samples with adjectival attachments and a low percentage of adverbial ones, or vice-versa. The optimal split would be such that all the subsets would contain only samples of one attachment type. For each attribute **A**, we split the set into subsets, each associated with attribute value $\mathbf{A_W}$ and containing samples which were unifiable with value $\mathbf{A_W}$ (belong to the same WordNet class). Then, we calculate the **overall heterogeneity (OH)** of all these subsets as a weighted sum of their expected information:

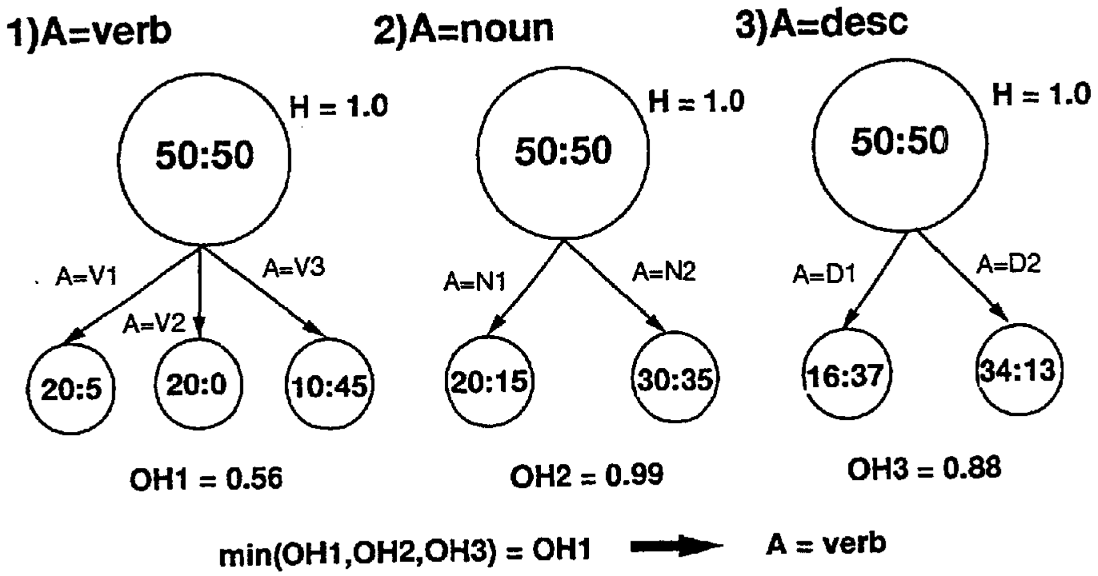
$$OH = -\sum_{w} p(A = A_w)[p(PP_{ADV} | A = A_w)\log_2 p(PP_{ADV} | A = A_w) + p(PP_{ADJ} | A = A_w)\log_2 p(PP_{ADJ} | A = A_w)],$$

where $p(PP_{ADV}|A{=}A_W)$ and $p(P_{ADJ}|A{=}A_W)$ represent the conditional probabilities of the adverbial and adjectival attachments, respectively. The attribute with the lowest overall heterogeneity is selected for the decision tree expansion. In the following example (Figure 2) we

[7] Classification in this case means deciding whether the PP is adjectival or averbial.

have to choose an attribute to split the node with 50 adjectival and 50 adverbial quadruples:

Figure 2: *Choosing an attribute for the decision tree expansion*



Verbs of all the node quadruples belong to the WordNet class V, nouns to the class N and descriptions to the class D. We assume, in this example, that the WordNet hierarchy class V has three subclasses (V1, V2, V3), class N has two subclasses (N1, N2) and class D has also two subclasses (D1, D2)[8]. We use the values V1, V2 and V3, N1 and N2, and D1 and D2 as potential values of the attribute A. Splitting by verb results in three subnodes with an overall heterogeneity 0.56, splitting by noun in two subnodes with OH=0.99 and by description with OH=0.88. Therefore, in this case we would choose the verb as an attribute for the tree expansion.

3. The attribute is either a verb, noun, or a description noun[9]. Its values correspond to the concept identificators (synsets) of WordNet. At the beginning of the tree induction, the top roots of the WordNet hierarchy are taken as attribute values for splitting the set of training examples. At first, all the training examples (separately for each preposition) are split into subsets which correspond to the topmost concepts of WordNet, which contains 11 topical roots for nouns and description nouns, and 337 for verbs (both nouns and verbs have hierarchical structure, although the hierarchy for verbs is shallower and wider). The training examples are grouped into subnodes according to the disambiguated senses of their content words. This means that quadruples with words that belong to the same top classes start at the same node. Each group is further split by the attribute, which provides less heterogeneous splitting (all verb, noun and description attributes are tried for each group and the one by which the current node can be split into the least heterogeneous set of subnodes is selected). Branches that lead to empty subnodes (as a result of not having a matching training example for the given attribute value) are pruned. This process repeats in all the emerging subnodes, using the attribute values which correspond to the WordNet hierarchy, moving from its top to its leaves. When splitting the set of training examples by the attribute A according to its values A_w, the emerging subsets contain those quadruples whose attribute A value is lower in the WordNet hierarchy, i.e. belongs to the same class. If some quadruples had the attribute value equal

[8] At the top of the tree we use all the roots of the WordNet hierarchy as initial subnodes.

[9] We induce the decision tree separately for each preposition.

to the values of A, an additional subset is added but its further splitting by the same attribute is prohibited.

3.2 CLASSIFICATION

As soon as the decision tree is induced, classifying an unseen quadruple is a relatively simple procedure. At first, the word senses of the quadruple are disambiguated by the algorithm described in Chapter 2, which is modified to exclude the SDT iteration cycles. Then a path is traversed in the decision tree, starting at its root and ending at a leaf. At each internal node, we follow the branch labelled by the attribute value which is the semantic ancestor of the attribute value of the quadruple (i.e. the branch attribute value is a semantic ancestor[10] of the value of the quadruple attribute). The quadruple is assigned the attachment type associated with the leaf, i.e. adjectival or adverbial. If no match is found for the attribute value of the quadruple at any given node, the quadruple is assigned the majority type of the current node.

4 TRAINING AND TESTING DATA

The training and testing data, extracted from the Penn Tree Bank [MA93], are identical to that used by [RRR94], [C&B95] for comparison purposes[11]. The data contained 20801 training and 3097 testing quadruples with 51 prepositions and ensured that there was no implicit training of the method on the test set itself. We have processed the training data in the following way:

- converted all the verbs into lower cases
- converted all the words into base forms
- replaced four digit numbers by 'year'
- replaced all other numbers by 'definite_quantity'
- replaced nouns ending by *-ing* and not in WordNet by 'action'
- eliminated examples with verbs that are not in WordNet
- eliminated examples with lower-case nouns that are not in WordNet, except for pronouns, whose senses were substituted by universal pronoun synsets
- the upper-case nouns were assigned their lower case equivalent senses plus the senses of 'company' and 'person'
- the upper case nouns not contained in WordNet were assigned the senses of 'company' and 'person'
- disabled all the intransitive senses of verbs
- assigned all the words (yet ambiguous) the sets of WordNet senses (synsets)

The above processing together with the elimination of double occurrences and contradicting examples, reduced the training set to 17577 quadruples, with an average quadruple ambiguity of 86, as of the ambiguity definition in section 1.2.

5. EVALUATION AND EXPERIMENTAL RESULTS

5.1 WORD SENSE DISAMBIGUATION

Because the induction of the decision tree for the PP attachment is based on a supervised learning from sense-tagged examples, it was necessary to sense-disambiguate the entire training set. This was done by the iterative algorithm described in Chapter 2.

[10] In the WordNet hierarchy.

[11] We would like to thank Michael Collins for supplying the data.

To form an approximate evaluation of the quality of this disambiguation, we have randomly selected 500 words, manually[12] assigned sets of possible senses to them (sets, because without a full sentential context a full disambiguation is not always possible), and compared these with the automatic disambiguation. If the automatically chosen sense was present in the manually assigned set, the disambiguation was considered correct. Out of these 500 words 362 could be considered correctly disambiguated, which represents slightly over 72%.

We can argue that the insufficient disambiguation context, sparse data problem and empirically set iteration step in the disambiguating algorithm lead to an unreliable disambiguation. However, it is necessary to maintain the understanding that it is the PP attachment rather than the sense disambiguation that is our primary goal. Additionally, because the words of the input sentences for the PP attachment are to be assigned senses in the same manner, the sense disambiguation error is concealed. Alhouhg the disambiguation of the training set is computationally the most expensive part of the system, it is done only once. The disambiguation of unseen (testing) examples is done by the same algorithm which is modified to exclude the SDT iteration cycles. It is therefore reasonably fast even for real-life applications.

5.2 PP-ATTACHMENT

The PP attachment using the decision tree is extremely efficient and reliable. We have induced the decision tree separately for each preposition in the training corpus, covering the 51 most common prepositions. The induced decision trees are relatively shallow and the classification of unseen sentences is rapid. As shown in the following table, our algorithm appears to have surpassed many existing methods and is very close to human performance on the same testing data[13].

TABLE 1: *PP Attachment Accuracy and comparison with other methods*

Method	**Percent Correct**
Always Adjectival	59.0
Most likely for each preposition	72.2
[RRR94]	81.6
[BR94] (different data)	81.8
[C&B95]	84.5
Induced decision tree	**88.1**
Average human (quadruple context only)	88.2
Induced decision tree (WordNet)[14]	90.8
Average human (whole sentence context)	93.2

The fact that many words in both the training and the testing sets were not found in WordNet caused a reduction in the accuracy. This is because training examples with an error or with a word not found in WordNet could not fully participate on the decision tree induction. This reduced the original training set of 20801 quadruples to 17577. In the case of the testing set, many of the 3097 testing quadruples were also handicapped by having no entry in WordNet. Attachment of these had to be based on a partial quadruple and was usually assigned at a higher level of the decision tree, which reduced the overall accuracy. In order to conduct a fair comparison, however, we used the same testing set as the methods shown in the above table. If just the examples with full WordNet entries were used, the accuracy rose to 90.8%.

[12] For the manual assignment we have used only the context of each quadruple plus the PP attachment information.

[13] We used different data preprocessing as described in Chapter 4.

[14] When tested only on the quadruples whose all words are found in WordNet.

Although the algorithm does not provide high enough accuracy from the point of view of word sense disambiguation, it is more important to bear in mind that our main goal is the PP attachment ambiguity resolution. The relatively low accuracy of the word sense disambiguation is compensated by the fact that the same sense disambiguation error is present in both the training set and the classified quadruple. The use of the same training set for both the PP attachment and the sense disambiguation provides a positive bias in favour of correct attachment.

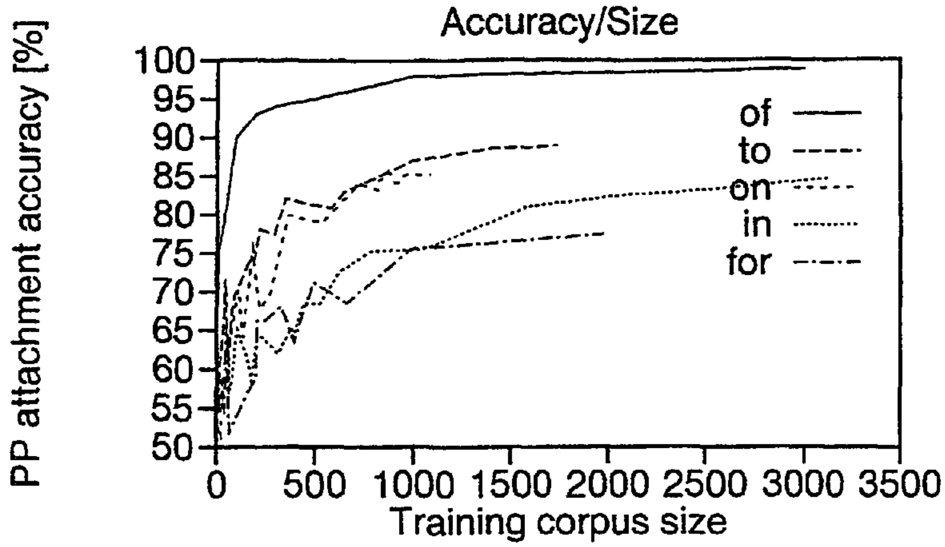
Until we have a sufficiently big enough word sense tagged corpus, we can only hypothesise on the importance of the correct sense disambiguation for the PP attachment. Experiments, however, show that if the positive bias between the word senses of the training set and the testing quadruples is removed, the accuracy of the PP attachment falls substantially. We have conducted an experiment, in which the disambiguated senses of the testing set were replaced by the most frequent senses, i.e. the first senses as defined in WordNet. This caused a substantial reduction of accuracy to 76.5%. The fact that our approximate disambiguation (algorithm in Chapter 2) leads to 88.1% correct PP attachment is partly to be attributed to the positive bias of disambiguation of the testing examples against the same training set which is also used for the decision tree induction. The disambiguation errors are thus hidden by their replication in both the training and the testing sets.

As we have already mentioned, Collins and Brooks [C&B95] based their method on matching the testing quadruples against the set of training examples. The decision on the attachment was made according to which attachment type had a higher count in the training corpus. If no match for the given quadruple was found, the algorithm backed-off to a combined frequency count of the occurences of matches on three words only, i.e. on the verb-noun-preposition, verb-preposition-description and noun-preposition-description. If no match was found on any of the three words combination, the algorithm backed-off to a combined match on two words, i.e. one of the content words with a preposition. If there was further no match found on two words, the attachment type was assigned according to the prepositional statistics, or, if the preposition was not present in the training corpus, the quadruple was assigned the adjectival default. There was a substantial decrease of accuracy between the triples and doubles stage. Our algorithm, on the other hand, has substantially reduced the number of classifications based on fewer words. This is because at the top of the decision tree all of the semantic tops of all of the content words of the given quadruple are compared with the semantic generalisations of the training examples represented through the nodes of the decision tree. Only if the homogeneity termination condition is satisfied before all three content words are compared, the decision is based on less than a full quadruple. The decision tree therefore represents a very useful mechanism for determining the semantic level at which the decision on the PP attachment is made.

Collins and Brooks' have also demonstrated the importance of low count events in training data by an experiment where all counts less than 5 were put to zero. This effectively made their algorithm ignore low count events which resulted in the decrease of accuracy from 84.1 to 81.6%. This important feature is maintained in our approach by small homogenous leaves at higher levels of the decision tree, which usually accommodate the low count training examples.

Figure 3 shows an interesting aspect of learning the prepositional phrase attachment from a huge corpus. We have selected five most common prepositions and compared their learning curves. It turned out that for the size of a training set smaller than 1000 examples, learning is rather unreliable and dependent on the quality of the chosen quadruples. For a bigger training set, the accuracy grows with its size until a certain maximum accuracy level is reached. This level is different for different prepositions and we hypothesise that it can be broken only when a wider sentential or discourse context is used.

Figure 3: *Accuracy/Corpus Size dependency curve*



Our algorithm also provides a qualification certainty based on the heterogeneity of the decision tree leaves. The tree leaves are heterogeneous for two reasons: 1) the tree expansion is terminated when a node contains more than 77% of examples belonging to the same class, or, 2) when there are examples in the node that cannot be further divided because the tree has reached the bottom of the WordNet hierarchy. The Table 2 shows that the incorrect attachments usually occur with a lower certainty than the correct ones, i.e. most of the incorrect attachments are marked as less certain.

TABLE 2: *Certainty evaluation*

Certainty	Number	Percent	Number Correct	Accuracy [%]
1.0	1424	46.0	1226	86.1
0.8 - 1.0	1261	40.7	1219	96.7
0.5 - 0.8	233	7.5	143	61.4
Prepositional statistics	176	5.7	137	77.8
Adjectival default	3	0.1	3	100.0
TOTALS	**3097**	**100.0**	**2728**	**88.1**

The *prepositional statistics* indicates that there were no matches found for the given quadruple and the attachment was decided based on the statistical frequency of the given preposition. *Adjectival default* was used in three cases when the preposition was not found in the training set. The *certainty between 0.5 and 0.8* accounts mostly for the examples whose attachment was made through the decision tree, but there was either a small number of examples that participated on the creation of the tree branch or the examples were not sufficiently representative (e.g. contradictory examples). Most of the examples in this category possibly require a wider sentential context for further improvement of accuracy. The *certainty bigger than 0.8 and smaller than 1.0* accounts for the situations when the decision was based on a leaf whose further expansion was terminated by the homogeneity termination condition or simply some noisy or incorrectly disambiguated examples were involved in its creation[15]. Examples, which did not reach the bottom of the decision tree and were assigned the majority class of the node from which there was no appropriate branch to follow, were all classified with certainty between 0.5 and 1.0. The decision with *certainty 1.0* is always based on a homogenous leaf. It does not exhibit the highest accuracy because many of the homogenous leaves are formed from only very few examples and many of these are erroneous.

As Figure 3 shows, each preposition has a different saturation accuracy which cannot be surpassed unless a wider sentential context is used. We believe, however, that a bigger corpus would provide better word-sense disambiguation which in turn would allow to increase the homogeneity limit for the termination of the tree expansion. Heterogeneous nodes, which force the expansion of the decision tree to unnecessary extent, are caused by 1) examples with an error in the word sense disambiguation, or by 2) examples, that can be both adjectival and adverbial if taken out of context. The second case cannot be eliminated by a bigger training corpus, however, the reduction of noisy examples would contribute to an increase in accuracy mainly in the case of small nodes which can now contain more noisy examples than correct ones and thus force a wrong attachment. We feel that a bigger corpus, would provide us with an increase of accuracy of "certainty 1" attachments, which partly includes attachments based on the small leaves. Also, we believe that a bigger training corpus would increase performance in the case of less frequent prepositions which do not have enough training examples to allow for induction of a reliable decision tree.

6. CONCLUSION AND FURTHER WORK

The most computationally expensive part of the system is the word sense disambiguation of the training corpus. This, however, is done only once and the disambiguated corpus is stored for future classifications of unseen quadruples. The above experiments confirmed the expectations that using the semantic information in combination with even a very limited context leads to a substantial improvement of NLP techniques. Although our method exhibits an accuracy close to the human performance, we feel that there is still a space for improvement, particularly in using a wider sentential context (human performance on full sentential context is over 93%), more training data and/or more accurate sense disambiguation technique. We believe that there is further space for elaboration of our method, in particular, it would be interesting to know the exact relations between the accuracy and the termination condition, and between the corpus size and the optimum termination condition separately for each preposition. At the moment, we are working on an implementation of the algorithm to work on with a wider sentential context and on its incorporation within a more complex NLP system.

REFERENCES

[A&S88] Altmann, G., Steedman, M., Interaction with Context During Human Sentence Processing. *Cognition, 30:191-238,*1988.

[B91] Basili, R., et al, Combining NLP and statistical techniques for lexical acquisition, *In Proceedings of the AAAI Fall Symposium on Probabilistic Apporaches to Natural Language, Computational Linguistics, 18(4): 467-480,* 1992.

[B&R94] Brill, E., Resnik, P., A Rule Based Approach to PP Attachment Disambiguation, *In Proceedings of COLING,* 1994.

[15] The relatively high accuracy of this certainty is due mostly to the preposition "of" which formes 29% of the testing data and whose decision tree is terminated at the very top of the expansion bacause of the homogenity termination condition (over 99% of the quadruples with the preposition "of" are adjectival).

[BR92] Brill, E., A Simple Rule-Based POS Tagger, *In Proceedings of the 3rd Conference on Applied Natural Language Processing*, 1992.

[BR93] Brill, E., Automatic Grammar Induction and Parsing Free Text: A Transformation Based Approach, *In Proceedings of the 31st Meeting of the ACL*, 1993.

[B&W94] Bruce, R., Wiebe, J., A new approach to word sense disambuiguation, *In Proceedings of the ARPA Workshop on Human Language Technology*, 1994.

[BR91] Bruha, I., Machine Learning: Empirical Methods. *In Proceedings of Sofsem 91*, 1991.

[C&B95] Collins, M., Brooks, J., PP Attachment Through a Backed-Off Model, *In Proceedings of the Third Workshop on Very Large Corpora*, 1995.

[GCY92] Gale, W., Church, K., Yarowsky, D., A method for disambiguating word senses in a large corpus, *Computers and Humanities*, 26, pp. 415-439,1992.

[H&R93] Hindle, H., Rooth, M.,. Structural Ambiguity and Lexical Relations, *Computational Linguistics, 19(1)103-120*, 1993.

[K&E96] Karov, Y., Edelman, S., Learning similarity-based word sense disambiguation, *Proceedings of the Fourth Workshop on Very Large Corpora*, pp. 42-55, 1996.

[KI73] Kimball, J., Seven Principles of Surface Structure Parsing in Natural Language, *Cognition, 2*, 1973.

[LI96] Li, H., A Probabilistic Disambiguation Method Based on Psycholinguistic Principles, *Proceedings of the Fourth Workshop on Very Large Corpora*, 1996

[MA93] Marcus, M., Building a Large Annotated Corpus of English: The Penn Treebank, *Association for Computational Linguistics*, 1993.

[MI90] Miller, G., Wordnet: an On-Line Lexical Database, *International Journal of Lexicography*, 1990

[MI93] Miller, G., Introduction to WordNet: An On-line Lexical Database, Princeton University. *ftp://clarity.princeton.edu*, 1993.

[MC94] Miller, G., et al. Using Semantic Concordance for Sense Identification, *In Proceedings of the Human Language Technology Workshop, p.240-243*, 1994.

[RRR94] Ratnaparkhi, A., Maximum Entrophy Model for PP Attachment, In *Procedings of the ARPA Workshop on Human Language Technology*,1994.

[RE95] Resnik, P., Disambiguating Noun Groupings with Respect to WordNet Senses, *In Proceedings of the Third Workshop on Very Large Corpora*, 1995.

[SU95] Sussna, M., Information Retrieval using Semantic Distance in WordNet, Personal communication, *sussna@cs.ucsd.edu*, 1995.

[SU96] Sussna, M., Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network, Personal communication, *sussna@cs.ucsd.edu*,1996.

[Q86] Quinlan, J., R., Induction of Decision Trees, *Machine Learning*, 1, pp 81-106, 1986.

[W91] Weischedel, R., et al, Partial Parsing: a report of work in progress, *In Proc. of the Fourth DARPA Speech and Natural Language Workshop*,1991.

[YA93] Yarowsky, D., One sense per collocation, *In Proceedings of the ARPA Workshop on Human Language Technology*, pp. 266-271, 1993.

[YA95] Yarowsky, D., Unsupervised word sense disambiguation rivalling supervised methods, *In Proceedings of the 33rd Annual Meeting of the ACL*, 1995.