Resolving PP attachment Ambiguities with Memory-Based Learning

Jakub Zavrel, Walter Daelemans, Jorn Veenstra
Computational Linguistics, Tilburg University
PO Box 90153, 5000 LE Tilburg, The Netherlands
zavrel,walter,veenstra@kub.nl

Abstract

In this paper we describe the application of Memory-Based Learning to the problem of Prepositional Phrase attachment disambiguation. We compare Memory-Based Learning, which stores examples in memory and generalizes by using intelligent similarity metrics, with a number of recently proposed statistical methods that are well suited to large numbers of features. We evaluate our methods on a common benchmark dataset and show that our method compares favorably to previous methods, and is well-suited to incorporating various unconventional representations of word patterns such as value difference metrics and Lexical Space.

1 Introduction

A central issue in natural language analysis is structural ambiguity resolution. A sentence is structurally ambiguous when it can be assigned more than one syntactic structure. The drosophila of structural ambiguity resolution is Prepositional Phrase (PP) attachment. Several sources of information can be used to resolve PP attachment ambiguity. Psycholinguistic theories have resulted in disambiguation strategies which use syntactic information only, i.e. structural properties of the parse tree are used to choose between different attachment sites. Two principles based on syntactic information are Minimal Attachment (MA) and Late Closure (LC) (Frazier, 1979). MA tries to construct the parse tree that has the fewest nodes, whereas LC tries to attach new constituents as low in the parse tree as possible. These strategies always choose the same attachment regardless of the lexical content of the sentence. This results in a wrong attachment in one of the following sentences:

1 *She eats pizza with a fork.*

2 *She eats pizza with anchovies.*

In sentence 1, the PP "with a fork" is attached to the verb "eats" (high attachment). Sentence 2 differs only minimally from the first sentence; here, the PP "with anchovies" does not attach to the verb but to the NP "pizza" (low attachment). In languages like English and Dutch, in which there is very little overt case marking, syntactic information alone does not suffice to explain the difference in attachment sites between such sentences. The use of syntactic principles makes it necessary to re-analyse the sentence, using semantic or even pragmatic information, to reach the correct decision. In the example sentences 1 and 2, the meaning of the head of the object of 'with' determines low or high attachment. Several semantic criteria have been worked out to resolve structural ambiguities. However, pinning down the semantic properties of all the words is laborious and expensive, and is only feasible in a very restricted domain. The modeling of pragmatic inference seems to be even more difficult in a computational system.

Due to the difficulties with the modeling of semantic strategies for ambiguity resolution, an attractive alternative is to look at the statistics of word patterns in annotated corpora. In such a corpus, different kinds of information used to resolve attachment ambiguity are, implicitly, represented in co-occurrence regularities. Several statistical techniques can use this information in learning attachment ambiguity resolution.

Hindle and Rooth (1993) were the first to show that a corpus-based approach to PP attachment ambiguity resolution can lead to good results. For sentences with a *verb/noun* attachment ambiguity, they measured the lexical association between the *noun* and the *preposition*, and the *verb* and the *preposition* in unambiguous sentences. Their method bases attachment decisions on the ratio and

Jakub Zavrel, Walter Daelemans and Jorn Veenstra (1997) **Resolving PP attachment Ambiguities with Memory-Based Learning.** In T.M. Ellison (ed.) *CoNLL97: Computational Natural Language Learning,* ACL pp 136-144.

reliability of these association strengths. Note that Hindle and Rooth did not include information about the second noun and therefore could not distinguish between sentence 1 and 2. Their method is also difficult to extend to more elaborate combinations of information sources.

More recently, a number of statistical methods better suited to larger numbers of features have been proposed for PP-attachment. Brill and Resnik (1994) applied Error-Driven Transformation-Based Learning, Ratnaparkhi, Reynar and Roukos (1994) applied a Maximum Entropy model, Franz (1996) used a Loglinear model, and Collins and Brooks (1995) obtained good results using a Back-Off model.

In this paper, we examine whether Memory-Based Learning (MBL), a family of statistical methods from the field of Machine Learning, can improve on the performance of previous approaches. Memory-Based Learning is described in Section 2. In order to make a fair comparison, we evaluated our methods on the common benchmark dataset first used in Ratnaparkhi, Reynar, and Roukos (1994). In section 3, the experiments with our method on this data are described. An important advantage of MBL is its use of *similarity-based reasoning*. This makes it suited to the use of various unconventional representations of word patterns (Section 2). In Section 3 a comparison is provided between two promising representational forms. Section 4 contains a comparison of our method to previous work, and we conclude with section 5.

2 Memory-Based Learning

Classification-based machine learning algorithms can be applied in learning disambiguation problems by providing them with a set of examples derived from an annotated corpus. Each example consists of an input vector representing the context of an attachment ambiguity in terms of features (e.g. syntactic features, words, or lexical features in the case of PP-attachment), and an output class (one of a finite number of possible attachment positions representing the correct attachment position for the input context). Machine learning algorithms extrapolate from the examples to new input cases, either by extracting regularities from the examples in the form of rules, decision trees, connection weights, or probabilities in greedy learning algorithms, or by a more direct use of analogy in lazy learning algorithms. It is the latter approach which we investigate in this paper. It is our experience that lazy learning (such as the Memory-Based Learning approach adopted here) is more effective for several language-processing problems (see Daelemans (1995) for an overview) than more eager learning approaches. Because language-processing tasks typically can only be described as a complex interaction of regularities, subregularities and (families of) exceptions, storing all empirical data as potentially useful in analogical extrapolation works better than extracting the main regularities and forgetting the individual examples (Daelemans, 1996).

Analogy from Nearest Neighbors

The techniques used are variants and extensions of the classic k-nearest neighbor (k-NN) classifier algorithm. The instances of a task are stored in a table, together with the associated "correct" output. When a new pattern is processed, the k nearest neighbors of the pattern are retrieved from memory using some similarity metric. The output is determined by extrapolation from the k nearest neighbors. The most common extrapolation method is *majority voting* which simply chooses the most common class among the k nearest neighbors as an output.

Similarity metrics

The most basic metric for patterns with symbolic features is the **Overlap metric** given in Equations 1 and 2; where $\Delta(X, Y)$ is the distance between patterns X and Y, represented by n features, w_i is a weight for feature i, and δ is the distance per feature. The k-NN algorithm with this metric, and equal weighting for all features is called IB1 (Aha, Kibler, and Albert, 1991). Usually k is set to 1.

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \; \delta(x_i, y_i) \tag{1}$$

where:

$$\delta(x_i, y_i) = 0 \; if \; x_i = y_i, \; else \; 1 \tag{2}$$

This metric simply counts the number of (mis)matching feature values in both patterns. If no information about the importance of features is available, this is a reasonable choice. But if we have information about feature relevance, we can add linguistic bias to weight or select different features (Cardie, 1996). An alternative, more empiricist, approach is to look at the behavior of features in the set of examples used for training. We can compute statistics about the relevance of features by looking at which features are good predictors of the class labels. Information Theory provides a useful tool for measuring feature relevance in this way, see Quinlan (1993).

Information Gain (IG) weighting looks at each feature in isolation, and measures how much information it contributes to our knowledge of the correct class label. The Information Gain of feature f is measured by computing the difference in uncertainty (i.e. entropy) between the situations without and with knowledge of the value of that feature (Equation 3):

$$w_f = H(C) - \frac{\sum_{v \in V_f} P(v) \times H(C|v)}{si(f)} \qquad (3)$$

$$si(f) = -\sum_{v \in V_f} P(v) \log_2 P(v) \qquad (4)$$

Where C is the set of class labels, V_f is the set of values for feature f, and $H(C) = -\sum_{c \in C} P(c) \log_2 P(c)$ is the entropy of the class labels. The probabilities are estimated from relative frequencies in the training set. The normalizing factor $si(f)$ (split info) is included to avoid a bias in favor of features with more values. It represents the amount of information needed to represent all values of the feature (Equation 4). The resulting IG values can then be used as weights in Equation 1. The k-NN algorithm with this metric is called IB1-IG, see Daelemans and van den Bosch (1992).

The possibility of automatically determining the relevance of features implies that many different and possibly irrelevant features can be added to the feature set. This is a very convenient methodology if theory does not constrain the choice sufficiently beforehand, or if we wish to measure the importance of various information sources experimentally.

MVDM and LexSpace

Although IB1-IG solves the problem of feature relevance to a certain extent, it does not take into account that the symbols used as values in the input vector features (in this case words, syntactic categories, etc.) are not all equally similar to each other. According to the Overlap metric, the words *Japan* and *China* are as similar as *Japan* and *pizza*. We would like *Japan* and *China* to be more similar to each other than *Japan* and *pizza*. This linguistic knowledge could be encoded into the word representations by hand, e.g. by replacing words with semantic labels, but again we prefer a more empiricist approach in which distances between values of the same feature are computed differentially on the basis of properties of the training set. To this end, we use the Modified Value Difference Metric (MVDM), see Cost and Salzberg (1993); a variant of a metric first defined in Stanfill and Waltz (1986). This metric (Equation 5) computes the frequency distribution of each value of a feature over the categories. Depending on the similarity of their distributions, pairs of values are assigned a distance.

$$\delta(V_1, V_2) = \sum_{i=1}^{n} |P(C_i|V_1) - P(C_i|V_2)| \qquad (5)$$

In this equation, V_1 and V_2 are two possible values for feature f; the distance is the sum over all n categories; and $P(C_i|V_j)$ is estimated by the relative frequency of the value V_j being classified as category i.

In our PP-attachment problem, the effect of this metric is that words (as feature values) are grouped according to the category distribution of the patterns they belong to. It is possible to cluster the distributions of the values over the categories, and obtain classes of similar words in this fashion. For an example of this type of unsupervised learning as a side-effect of supervised learning, see Daelemans, Berck, and Gillis (1996). In a sense, the MVDM can be interpreted as implicitly implementing a statistically induced, distributed, non-symbolic representation of the words. In this case, the category distribution for a specific word is its lexical representation. Note that the representation for each word is entirely dependent on its behavior with respect to a particular classification task.

In many practical applications of MB-NLP, we are confronted with a very limited set of examples. This poses a serious problem for the MVD metric. Many values occur only once in the whole data set. This means that if two such values occur with the same class, the MVDM will regard them as identical, and if they occur with two different classes their distance will be maximal. In many cases, the latter condition reduces the MVDM to the overlap metric, and additionaly some cases will be counted as an exact match on the basis of very shaky evidence. It is, therefore, worthwile to investigate whether the value difference matrix $\delta(V_i, V_j)$ can be reused from one task to another. This would make it possible to reliably estimate all the δ parameters on a task for which we have a large amount of training material, and to profit from their availability for the MVDM of a smaller domain.

Such a possibility of reuse of lexical similarity is found in the application of Lexical Space representation (Schütze, 1994, Zavrel and Veenstra, 1995). In LexSpace, each word is represented by a vector of real numbers that stands for a "fingerprint" of the words' distributional behavior across local contexts in a large corpus. The distances between vectors can be taken as a measure of similarity. In Table 1, a number of examples of nearest neighbors are shown.

For each focus-word f, a score is kept of the number of co-occurrences of words from a fixed set of C context-words w_i $(1 < i < C)$ in a large corpus. Previous work by Hughes (1994) indicates that the two neighbors on the left and on the right (i.e. the words in positions $n-2$, $n-1$, $n+1$, $n+2$, relative to word n) are a good choice of context. The position of a word in Lexical Space is thus given by a four component vector, of which each component has as many dimensions as there are context words. The dimensions represent the conditional probabilities $P(w_1^{n-2}|f) \ldots P(w_c^{n+2}|f)$.

We derived the distributional vectors of all 71479 unique words present in the 3 million words of Wall Street Journal text, taken from the ACL/DCI CD-ROM I (1991). For the contexts, i.e. the dimensions of Lexical Space, we took the 250 most frequent words.

To reduce the 1000 dimensional Lexical Space vectors to a manageable format we applied Principal Component Analysis[1] (PCA) to reduce them to a much lower number of dimensions. PCA accomplishes the dimension reduction that preserves as much of the structure of the original data as possible. Using a measure of the correctness of the classification of a word in Lexical Space with respect to a linguistic categorization (see Zavrel and Veenstra (1995)) we found that PCA can reduce the dimensionality from 1000 to as few as 25 dimensions with virtually no loss, and sometimes even an improvement of the quality of the organization.

Note that the LexSpace representations are task independent in that they only reflect the structure of neighborhood relations between words in text. However, if the task at hand has some positive relation to context prediction, Lexical Space representations are useful.

3 MBL for PP attachment

This section describes experiments with a number of Memory-Based models for PP attachment disambiguation. The first model is based on the lexical information only, i.e. the attachment decision is made by looking only at the identity of the words in the pattern. The second model considers the issue of lexical representation in the MBL framework, by taking as features either task dependent (MVDM) or task independent (LexSpace) syntactic vector representations for words. The introduction of vector represenations leads to a number of modifications to the distance metrics and extrapolation rules in the MBL framework. A final experiment examines a number of weighted voting rules.

The experiments in this section are conducted on a simplified version of the "full" PP-attachment problem, i.e. the attachment of a PP in the sequence: `VP NP PP`. The data consist of four-tuples of words, extracted from the Wall Street Journal Treebank (Marcus, Santorini, and Marcinkiewicz, 1993) by a group at IBM (Ratnaparkhi, Reynar, and Roukos, 1994).[2] They took all sentences that contained the pattern `VP NP PP` and extracted the head words from the constituents, yielding a `V N1 P N2` pattern. For each pattern they recorded whether the PP was attached to the verb or to the noun in the treebank parse. Example sentences 1 and 2 would then become:

`3 eats, pizza, with, fork, V.`

`4 eats, pizza, with, anchovies, N.`

The data set contains 20801 training patterns, 3097 test patterns, and an independent validation set of 4039 patterns for parameter optimization. It has been used in statistical disambiguation methods by Ratnaparkhi, Reynar, and Roukos (1994) and Collins and Brooks (1995); this allows a comparison of our models to the methods they tested. All of the models described below were trained on all of the training examples and the results are given for the 3097 test patterns. For the benchmark comparison with other methods from the literature, we use only results for which all parameters have been optimized on the validation set.

In addition to the computational work, Ratnaparkhi, Reynar, and Roukos (1994) performed a study with three human subjects, all experienced treebank annotators, who were given a small random sample of the test sentences (either as four-tuples or as full sentences), and who had to give the same binary decision. The humans, when given the four-tuple, gave the same answer as the Treebank parse 88.2 % of the time, and when given the whole sentence, 93.2 % of the time. As a baseline, we can consider either the Late Closure principle, which always attaches to the noun and yields a score of only

[1] Using the `simplesvd` package, which was kindly provided by Hinrich Schütze. This software can be obtained from `ftp://csli.stanford.edu /pub/prosit/papers/simplesvd/`.

[2] The dataset is avaliable from `ftp://ftp.cis.upenn.edu/pub/adwait/PPattachData/`. We would like to thank Michael Collins for pointing this benchmark out to us.

IN	*in*			
for(in)0.05	since(in)0.10	at(in)0.11	after(in)0.11	under(in)0.11
on(in)0.12	until(in)0.12	by(in)0.13	among(in)0.14	before(in)0.16
GROUP	*nn*			
network(nn)0.08	firm(nn)0.11	measure(nn)0.11	package(nn)0.11	chain(nn)0.11
club(np)0.11	bill(nn)0.11	partnership(nn)0.12	panel(nn)0.12	fund(nn)0.12
JAPAN	*np*			
china(np)0.16	france(np)0.16	britain(np)0.19	canada(np)0.19	mexico(np)0.19
india(np)0.19	australia(np)0.20	korea(np)0.22	italy(np)0.23	detroit(np)0.23

Table 1: Some examples of the direct neighbors of words in a Lexical Space (context:250 lexicon:5000 norm:1). The 10 nearest neighbors of the word in upper case are listed by ascending distance.

59.0 % correct, or the most likely attachment associated with the preposition, which reaches an accuracy of 72.2 %.

The training data for this task are rather sparse. Of the 3097 test patterns, only 150 (4.8 %) occurred in the training set; 791 (25.5 %) patterns had at least 1 mismatching word with any pattern in the training set; 1963 (63.4 %) patterns at least 2 mismatches; and 193 (6.2 %) patterns at least 3 mismatches. Moreover, the test set contains many words that are not present in any of the patterns in the training set. Table 2 shows the counts of feature values and unknown values. This table also gives the Information Gain estimates of feature relevance.

Overlap-Based Models

In a first experiment, we used the IB1 algorithm and the IB1-IG algorithm. The results of these algorithms and other methods from the literature are given in Table 3. The addition of IG weights clearly helps, as the high weight of the P feature in effect penalizes the retrieval of patterns which do not match in the preposition. As we have argued in Zavrel and Daelemans (1997), this corresponds exactly to the behavior of the Back-Off algorithm of Collins and Brooks (1995), so that it comes as no surprise that the accuracy of both methods is the same. Note that the Back-Off model was constructed after performing a number of validation experiments on held-out data to determine which terms to include and, more importantly, which to exclude from the back-off sequence. This process is much more laborious than the automatic computation of IG-weights on the training set.

The other methods for which results have been reported on this dataset include decision trees, Maximum Entropy (Ratnaparkhi, Reynar, and Roukos, 1994), and Error-Driven Transformation-Based Learning (Brill and Resnik, 1994),[3] which were clearly outperformed by both IB1 and IB1-IG, even though e.g. Brill & Resnik used more elaborate feature sets (words and WordNet classes). Adding more elaborate features is also possible in the MBL framework. In this paper, however, we focus on more effective use of the existing features. Because the Overlap metric neglects information about the degree of mismatch if feature-values are not identical, it is worthwhile to look at more finegrained representations and metrics.

[3]The results of Brill's method on the present benchmark were reconstructed by Collins and Brooks (1995).

Continuous Vector Representations for Words

In experiments with Lexical Space representations, every word in a pattern was replaced by its PCA compressed LexSpace vector, yielding patterns with 25x4 numerical features and a discrete target category. The distance metric used was the sum of the LexSpace vector distance per feature, where the distance between two vectors is computed as one minus the cosine, normalized by the cumulative norm. Because no two patterns have the same distance in this case, to use only the nearest neighbor(s) means extrapolating from exactly one nearest neighbor.

In preliminary experiments, this was found to give bad results, so we also experimented with various settings for k : the parameter that determines the number of neighbors considered for the analogy. The same was done for the MVDM metric which has a similar behavior. We found that LexSpace performed best when k was set to 13 (83.3 % correct); MVDM obtained its best score when k was set to 50 (80.5 % correct). Although these parameters were found by optimization on the test set, we can see in Figure 1 that LexSpace actually outperforms MVDM for all settings of k. Thus, the representations from LexSpace which represent the behavior of the values independent of the requirements of this particular classification task outperform the task specific representations used by MVDM. The reason is that the task specific representations are derived only from the small number of occurrences of each value in the training set, whereas the amount of text available to refine the LexSpace vectors is practically unlimited. Lexical Space however, does not outperform the simple Overlap metric (83.7 % correct) in this form. We suspected that the reason for this is the fact that when continuous represen-

Feature	train values	total values	unknown	IG weight
V	3243	3475	232	0.03
N1	4315	4613	298	0.03
P	66	69	3	0.10
N2	5451	5781	330	0.03
C	2	2	0	–

Table 2: Statistics of the PP attachment data set.

Method	percent correct
Overlap	83.7 %
Overlap IG ratio	84.1 %
C4.5	79.7 %
Maximum Entropy	77.7 %
Transformations	81.9 %
Back-off model	84.1 %
Late Closure	59.0 %
Most Likely for each P	72.0 %

Table 3: Scores on the Ratnaparkhi et al. PP-attachment test set (see text); the scores of Maximum Entropy are taken from Ratnaparkhi et al. (1994); the scores of Transformations and Back-off are taken from Collins & Brooks (1995). The C4.5 decision tree results, and the baselines have been computed by the authors.

tations are used, the number of neighbors is exactly fixed to k, whereas the number of neighbors used in the Overlap metric is, in effect, dependent on the specificity of the match.

Weighted Voting

This section examines possibilities for improving the behavior of LexSpace vectors for MBL by considering various *weighted voting* methods.

The fixed number of neighbors in the continuous metrics can result in an *oversmoothing* effect. The k-NN classifier tries to estimate the conditional class probabilities from samples in a local region of the data space. The radius of the region is determined by the distance of the k-furthest neighbor. If k is very small and i) the nearest neighbors are not nearby due to data sparseness, or ii) the nearest neighbor classes are unreliable due to noise, the "local" estimate tends to be very poor, as illustrated in Figure 1. Increasing k and thus taking into account a larger region around the query in the dataspace makes it possible to overcome this effect by smoothing the estimate. However, when the majority voting method is used, smoothing can easily become oversmoothing, because the radius of the neighborhood is as large as the distance of the k'th nearest neighbor, irrespective of the local properties of the data. Selected points from beyond the "relevant neighborhood" will receive a weight equal to the close neighbors in the voting function, which can result in unnecessary classification errors.

A solution to this problem is the use of a weighted voting rule which weights the vote of each of the nearest neighbors by a function of their distance to the test pattern (query). This type of voting rule was first proposed by Dudani (1976). In his scheme, the nearest neighbor gets a weight of 1, the furthest neighbor a weight of 0, and the other weights are scaled linearly to the interval in between.

$$w_j = \begin{cases} \frac{d_k - d_j}{d_k - d_1} & \text{if } d_k \neq d_1 \\ 1 & \text{if } d_k = d_1 \end{cases} \tag{6}$$

where d_j is the distance to the query of the j'th nearest neighbor, d_1 the distance of the nearest neighbor, and d_k the distance of the furthest (k'th) neighbor.

Dudani further proposed the *inverse distance weight* (Equation 7), which has recently become popular in the MBL literature (Wettschereck, 1994). In Equation 7, a small constant is usually added to the denominator to avoid division by zero.

$$w_j = \frac{1}{d_j} \tag{7}$$

Another weighting function considered here is based on the work of Shepard (1987), who argues for a universal perceptual law, in which the relevance of a previous stimulus for the generalization to a new stimulus is an exponentially decreasing function of its distance in a psychological space. This gives the weighed voting function of Equation 8, where α and β are constants determining the slope and the power of the exponential decay function. In the experiments reported below, $\alpha = 3.0$ and $\beta = 1.0$.

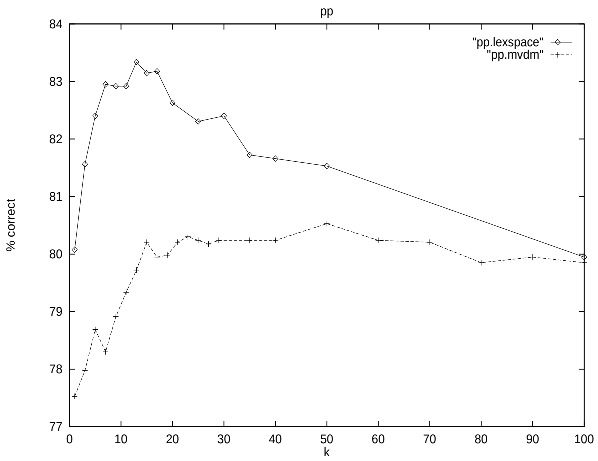

Figure 1: Accuracy on the PP-attachment test set of of MVDM and LexSpace representations as a function of k, the number of nearest neighbors.

Method	% correct
LexSpace (Dudani, k=30)	84.2 %
LexSpace (Dudani, k=50, IG)	84.4 %

Table 4: Scores on the Ratnaparkhi et al. PP-attachment test set with Lexical Space representations. The values of k, the voting function, and the IG weights were determined on the training and validation sets.

$$w_j = e^{-\alpha d_j^{\beta}} \tag{8}$$

Figure 2 shows the results on the test set for a wide range of k for these voting methods when applied to the LexSpace represented PP-attachment dataset.

With the inverse distance weighting function the results are better than with majority voting, but here, too, we see a steep drop for k's larger than 17. Using Dudani's weighting function, the results become optimal for larger values of k, and remain good for a wide range of k values. Dudani's weighting function also gives us the best overall result, i.e. if we use the best possible setting for k for each method, as determined by performance on the validation set (see Table 4).

The Dudani weighted k-nearest neighbor classifier (k=30) slightly outperforms Collins & Brooks' (1995) Back-Off model. A futher small increase was obtained by combining LexSpace representations with IG weighting of the features, and Dudani's weighted voting function. Although the improvement over Back-Off is quite limited, these results are nonetheless interesting because they show that MBL can gain from the introduction of extra information sources, whereas this is very difficult in the Back-Off algorithm. For comparison, consider that the performance of the Maximum Entropy model with distributional word-class features is still only 81.6% on this data.

4 Discussion

If we compare the accuracy of humans on the V,N,P,N patterns (88.2 % correct) with that of our most accurate method (84.4 %), we see that the paradigm of learning disambiguation methods from corpus statistics offers good prospects for an effective solution to the problem. After the initial effort by Hindle and Rooth (1993), it has become clear that this area needs statistical methods in which an easy integration of many information sources is possible. A number of methods have been applied to the task with this goal in mind.

Brill and Resnik (1994) applied Error-Driven Transformation-Based Learning to this task, using the *verb*, *noun*1, *preposition*, and *noun*2 features. Their method tries to maximize accuracy with a minimal amount of rules. They found an increase in performance by using semantic information from WordNet. Ratnaparkhi, Reynar, and Roukos (1994) used a Maximum Entropy model and a decision tree on the dataset they extracted from the Wall Street Journal corpus. They also report performance gains with word features derived by an unsupervised clustering method. Ratnaparkhi et al. ignored low frequency events. The accuracy of these two approaches is not optimal. This is most likely due to the fact that they treat low frequency events as noise, though these contain a lot of information in a sparse domain such as PP-attachment. Franz (1996) used a Loglinear model for PP attachment. The features he used were the preposition, the verb level

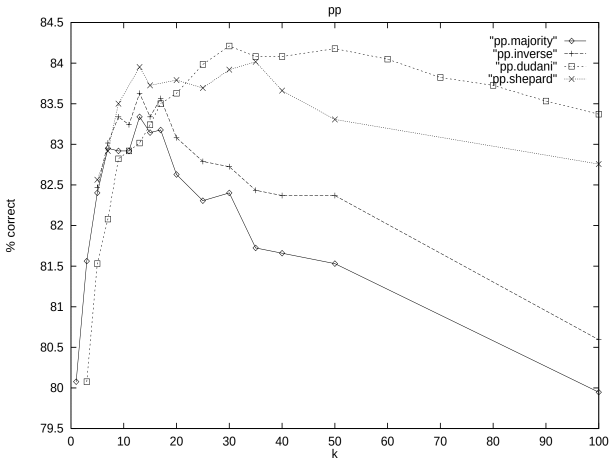

Figure 2: Accuracy on the PP-attachment test set of various voting methods as a function of k, the number of nearest neighbors.

(the lexical association between the *verb* and the *preposition*), the noun level (idem dito for *noun*1), the noun tag (POS-tag for *noun*1), noun definiteness (of *noun*1), and the PP-object tag (POS-tag for *noun*2). A Loglinear model keeps track of the interaction between all the features, though at a fairly high computational cost. The dataset that was used in Franz' work is no longer available, making a direct comparison of the performance impossible. Collins and Brooks (1995) used a Back-Off model, which enables them to take low frequency effects into account on the Ratnaparkhi dataset (with good results). In Zavrel and Daelemans (1997) it is shown that Memory-Based and Back-Off type methods are closely related, which is mirrored in the performance levels. Collins and Brooks got slightly better results (84.5 %) after reducing the sparse data problem by preprocessing the dataset, e.g. replacing all four-digit words with 'YEAR'. The experiments with Lexical Space representations have as yet not shown impressive performance gains over Back-Off, but they have demonstrated that the MBL framework is well-suited to experimentation with rich lexical representations.

5 Conclusion

We have shown that our MBL approach is very competent in solving attachment ambiguities; it achieves better generalization performance than many previous statistical approaches. Moreover, because we can measure the relevance of the features using an information gain metric (IB1-IG), we are able to add features without a high cost in model selection or an explosion in the number of parameters.

An additional advantage of the MBL approach is that, in contrast to the other statistical approaches, it is founded in the use of similarity-based reasoning. Therefore, it makes it possible to experiment with different types of distributed non-symbolic lexical representations extracted from corpora using unsupervised learning. This promises to be a rich source of extra information. We have also shown that task specific similarity metrics such as MVDM are sensitive to the sparse data problem. LexSpace is less sensitive to this problem because of the large amount of data which is available for its training.

Acknowledgements

This research was done in the context of the "Induction of Linguistic Knowledge" research programme, partially supported by the Foundation for Language Speech and Logic (TSL), which is funded by the Netherlands Organization for Scientific Research (NWO).

References

Aha, D., D. Kibler, and M. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37–66.

Brill, E. and P. Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proc. of 15th annual conference on Computational Linguistics.*

Cardie, Claire. 1996. Automatic feature set selection for case-based learning of linguistic knowledge. In *Proc. of Conference on Empirical Methods in NLP*. University of Pennsylvania.

Collins, M.J and J. Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Proc. of Third Workshop on Very Large Corpora*, Cambridge.

Cost, S. and S. Salzberg. 1993. A weighted nearest neighbour algorithm for learning with symbolic features. *Machine Learning*, 10:57–78.

Daelemans, W. 1995. Memory-based lexical acquisition and processing. In P. Steffens, editor, *Machine Translation and the Lexicon*, volume 898 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, pages 85–98.

Daelemans, W. 1996. Abstraction considered harmful: Lazy learning of language processing. In *Proc. of 6th Belgian-Dutch Conference on Machine Learning*, pages 3–12. Benelearn.

Daelemans, Walter, Peter Berck, and Steven Gillis. 1996. Unsupervised discovery of phonological categories through supervised learning of morphological rules. In *Proc. of 16th Int. Conf. on Computational Linguistics*, pages 95–100. Center for Sprogteknologi.

Daelemans, Walter and Antal van den Bosch. 1992. Generalisation performance of backpropagation learning on a syllabification task. In *Proc. of TWLT3: Connectionism and NLP*, pages 27–37. Twente University.

Dudani, S.A. 1976. The distance-weighted k-nearest neighbor rule. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume SMC-6, pages 325–327.

Franz, A. 1996. Learning PP attachment from corpus statistics. In S. Wermter, E. Riloff, and G. Scheler, editors, *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, volume 1040 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, New York, pages 188–202.

Frazier, L. 1979. On Comprehending Sentences: Syntactic Parsing Strategies. Ph.d thesis, University of Connecticut.

Hindle, D. and M. Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19:103–120.

Hughes, J. 1994. Automatically Acquiring a Classification of Words. Ph.d thesis, School of Computer Studies, The University of Leeds.

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

Quinlan, J.R. 1993. *c4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Ratnaparkhi, A., J. Reynar, and S. Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Workshop on Human Language Technology*, Plainsboro, NJ, March. ARPA.

Schütze, H. 1994. Distributional part-of-speech tagging. In *Proc. of 7th Conference of the European Chapter of the Association for Computational Linguistics*, Dublin, Ireland.

Shepard, R.N. 1987. Toward a universal law of generalization for psychological science. *Science*, 237:1317–1228.

Stanfill, C. and D. Waltz. 1986. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, December.

Wettschereck, D. 1994. A study of distance-based machine learning algorithms. Ph.d thesis, Oregon State University.

Zavrel, J. and W. Daelemans. 1997. Memory-based learning: Using similarity for smoothing. In *Proc. of 35th annual meeting of the ACL*, Madrid.

Zavrel, J. and J. Veenstra. 1995. The language environment and syntactic word class acquisition. In F. Wijnen and C. Koster, editors, *Proc. of Groningen Assembly on Language Acquisition (GALA95)*, Groningen.