Learning PP attachment for filtering prosodic phrasing

Olga van Herwijnen and **Jacques Terken**
Technology Management
Eindhoven University of Technology
P.O. Box 513, NL-5600 MB Eindhoven
The Netherlands
{O.M.v.Herwijnen,J.M.B.Terken}@tue.nl

Antal van den Bosch and **Erwin Marsi**
ILK / Comp. Ling. and AI
Tilburg University
P.O. Box 90153, NL-5000 LE Tilburg
The Netherlands
{A.vdnBosch,E.Marsi}@uvt.nl

Abstract

We explore learning prepositional-phrase attachment in Dutch, to use it as a filter in prosodic phrasing. From a syntactic treebank of spoken Dutch we extract instances of the attachment of prepositional phrases to either a governing verb or noun. Using cross-validated parameter and feature selection, we train two learning algorithms, IB1 and RIPPER, on making this distinction, based on unigram and bigram lexical features and a cooccurrence feature derived from WWW counts. We optimize the learning on noun attachment, since in a second stage we use the attachment decision for blocking the incorrect placement of phrase boundaries before prepositional phrases attached to the preceding noun. On noun attachment, IB1 attains an F-score of 82; RIPPER an F-score of 78. When used as a filter for prosodic phrasing, using attachment decisions from IB1 yields the best improvement on precision (by six points to 71) on phrase boundary placement.

1 Introduction

One of the factors determining the acceptability of synthetic speech is the appropriate placement of phrase boundaries, realized typically and most audibly by pauses (Sanderman, 1996). Incorrect prosodic phrasing may impede the listener in the correct understanding of the spoken utterance (Sanderman and Collier, 1997). A major factor causing difficulties in appropriate phrase boundary placement is the lack of reliable information about syntactic structure. Even if there is no one-to-one mapping between syntax and prosody, the placement of prosodic phrase boundaries is nevertheless dependent on syntactic information (Selkirk, 1984; Bear and Price, 1990; van Herwijnen and Terken, 2001b). To cope with this lack of syntactic information that a speech synthesis developer may face currently, e.g. in the absence of a reliable parser, several strategies have been applied to allocate phrase boundaries. One strategy is to allocate phrase boundaries on the basis of punctuation only. In general, however, this results in too few phrase boundaries (and some incorrect ones, e.g. in enumerations).

A clear example of information about syntactic structure being useful for placing phrase boundaries is the attachment of prepositional phrases (PPs). When a PP is attached to the preceding NP or PP (henceforth referred to as noun attachment), such as in the structure ... *eats pizza with anchovies*, a phrase boundary between *pizza* and *with* is usually considered inappropriate. However, when a PP is attached to the verb in the clause (verb attachment), as in the structure ... *eats pizza with a fork*, an intervening phrase boundary between the PP and its preceding NP or PP (between *pizza* and *with*) is optional, and when placed, usually judged appropriate (Marsi et al., 1997).

Deciding about noun versus verb attachment of PPs is a known hard task in parsing, since it is understood to involve knowing lexical preferences, verb subcategorization, fixed phrases, but also semantic and pragmatic "world" knowledge. A typical current parser (e.g., statistical parsers such as (Collins, 1996; Ratnaparkhi, 1997; Charniak, 2000)) interleaves PP attachment with all its other disambiguation tasks. However, because of its interesting complexity, a line of work has concentrated on studying the task in isolation (Hindle and Rooth, 1993; Ratnaparkhi et al., 1994; Brill and Resnik, 1994; Collins and Brooks, 1995; Franz, 1996; Zavrel et al., 1997). Our study can be seen as following these lines of isolation studies, pursuing the same process for another language, Dutch. At present there are no parsers available for Dutch that disambiguate PP attachment, which leaves the comparison between PP attachment as an embedded subtask of a full parser with our approach as future work.

In line with these earlier studies, we assume that at least two sources of information should be used as features in training data: (i) lexical features (e.g. unigrams and bigrams of head words), and (ii) word cooccurrence strength values (the probability that two words occur together, within some defined vicinity). Lexical features may be informative when certain individual words or bigrams frequently, or exclusively, occur with either noun or verb attachment. This may hold for prepositions, but also heads of the involved phrases, as well as for combinations of these words. Cooccurrence strength values may provide additional clues to informational ties among words; when we investigate the cooccurrences of nouns and prepositions, and of verbs and prepositions, the cooccurrence strength value could also indicate whether the prepositional phrase is attached to the noun or to the verb in the syntactic tree.

In this study, we use two machine learning algorithms to perform PP attachment. In line with the case study for English introduced in Ratnaparkhi et al. (1994), we collect a training set of Dutch PP attachment instances from a syntactic treebank. Collection of this data is described in Section 2. We extract lexical head features (unigram and bigram) from the treebank occurrences, and enrich this data with cooccurrence information extracted from the WWW (Section 3). Using the same features, we analogously build a held-out test corpus for which prosodic labeling is available. The setup of the machine learning experiments, involving automatic parameter and feature selection, is described in Section 4. We give the results of the cross-validation experiments on the original data and on the held-out data in Section 5. Employing the learned PP attachment modules for filtering phrase break placement is discussed in Section 6, where we test on the held-out written text corpus. We discuss our findings in Section 7.

2 Selection of material

From the Corpus Gesproken Nederlands (CGN, Spoken Dutch Corpus)[1], development release 5, we manually selected 1004 phrases that contain [NP PP] or [PP PP] sequences. Annotated according to protocol (van der Wouden et al., 2002), all PPs have been classified into noun or verb attachment. This classification yields 398 phrases (40%) with a verb-attached PP and 606 phrases (60%) with a noun-attached PP.

Additionally, as held-out corpus for testing the efficacy of PP attachment information for prosodic phrasing, we selected 157 sentences from various newspaper articles and e-mail messages. We selected this corpus because part of it had been annotated earlier on prosodic phrasing through a consensus transcription of ten phonetic experts (van Herwijnen and Terken, 2001a). All selected 157 sentences contain either [NP PP] or [PP PP] sequences. To obtain a "gold standard" we manually classified all PPs into NOUN and VERB attachment, according to the "single constituent test" (Paardekooper, 1977). This test states that every string of words that can be placed at the start of a finite main clause, forms a single constituent. Thus, if and only if a [NP PP] or [PP PP] sequence can be fronted, it forms a single NP containing a noun-attached PP. This classification resulted in 66 phrases with a verb-attached PP and 91 phrases with a noun-attached PP.

[1] The *Spoken Dutch Corpus* is a database of contemporary Dutch as spoken by adults in the Netherlands and Flanders. The project is funded by the Flemish and Dutch governments and the Netherlands Organization for Scientific Research NWO. Its homepage is `http://lands.let.kun.nl/cgn/ehome.htm`.

3 Feature engineering

3.1 Lexical features

Analogous to Ratnaparkhi et al. (1994), we (manually) selected the four lexical heads of the phrases involved in the attachment as features. We used the manually annotated phrasing and function labelling to determine the heads of all involved phrases. First, the noun of the preceding NP or PP that the focus PP might be attached to (N1); second, the preposition (P) of the PP to be attached; third, the verbal head (V) of the clause that the PP is in; and fourth, the noun head of the PP to be attached. For example, the Dutch sequence ... *[PP met Duits] [PP om de oren] [VP slaan]* (blow someone up over German), N1 is *Duits*, P is *om*, V is *slaan*, and N2 is *oren*. In the fixed expression *om de oren slaan*, *om de oren* attaches to *slaan*.

Subsequently, we added all combinations of two heads as features[2]. There are six possible combinations of the four heads: N1-P, N1-V, ... The example construction is thus stored in the data set as the following comma-separated 10-feature instance labelled with the VERB attachment class:

```
Duits, om, slaan, oren, Duits-om,
Duits-slaan, Duits-oren, om-slaan,
om-oren, slaan-oren, VERB
```

3.2 Cooccurrence strength values

Several metrics are available that estimate to what extent words or phrases belong together informationally. Well known examples of such cooccurrence strength metrics are mutual information (Church and Hanks, 1991), chi-square and log likelihood (Dunning, 1993). Cooccurrence strength values are typically estimated from a very large corpus. Often, these corpora are static and do not contain neologisms and names from later periods. In this paper, we explore an alternative by estimating cooccurrence strength values from the WWW. The WWW can be seen as a dynamic corpus: it contains new words that are not yet incorporated in other (static) corpora. Another advantage of using the WWW as a corpus is that it is the largest freely and electronically accessible corpus (for most languages including Dutch). Consequently, frequency counts obtained from the WWW are likely to be much more robust than those obtained from smaller corpora. If cooccurrences correlate with PP attachment, then the WWW could be an interesting robust background source of information. Recently, this reasoning was introduced in (Volk, 2000), a study in which the WWW was used to resolve PP attachment. Following this, the second step in engineering our feature set was to add cooccurrence strength values for Dutch words extracted from the WWW.

We explored three methods in which the cooccurrence strength value was used to decide between noun or verb attachment for all 1004 phrases from the CGN. The first method is a replication of the study by Volk (2000). In this study cooccurrence strength values were computed for the verb within close vicinity of the preposition Cooc(VnearP) and for the noun within close vicinity of the preposition Cooc(NnearP). Second, we investigated the method in which only Cooc(NnearP) is used. Third, we tested a variant on the second method by computing the cooccurrence strength value of a noun immediately succeeded by a preposition Cooc(N P), because there cannot be a word in between. The general formula for computing the cooccurrence strength value[3] of two terms is given by function (1) as proposed by Volk (2000). This method is based on the respective frequency of X and the joint frequency of X with a given preposition; where P stands for Preposition and X can be either a Noun or a Verb.

$$cooc(X\ P) = \frac{freq(X\ P)}{freq(X)} \tag{1}$$

We restricted the search to documents which were automatically identified as being written in Dutch by Altavista. For the Cooc(VnearP) and Cooc(NnearP) we used the advanced search function NEAR of the WWW search engine Altavista (Altavista, 2002). This function restricts the search to the appearance of two designated words at a maximal distance of 10 words, which is the default. The search is performed for both possible orders of appearance of the two desig-

[2] Note that Ratnaparkhi et al. (1994) allow all combinations of one to four heads as features.

[3] The notion cooccurrence strength value could also be referred to as relative frequency estimate of the conditional probability that a preposition co-occurs with a certain noun or verb.

Table 1: *Performance on PP attachment based on three variants of cooccurrence values.*

		NOUN attachment			VERB attachment		
	accuracy	precision	recall	$F_{\beta=1}$	precision	recall	$F_{\beta=1}$
NnearP or VnearP	62	71	62	66	51	61	56
NnearP	64	75	61	67	54	71	61
N P	67	84	54	65	55	87	67
baseline	60	60	100	75	-	0	-

nated words. For the Cooc(N P) we used the search function to search for exact multi-word phrases: "<noun> <prep>". This function restricts the search to the appearance of the two adjacent words in the indicated order. The number of found documents according to these search methods was used for *freq(X P)*. The *freq(X)* was derived from the WWW by performing a separate search for the single word form.

Method I: cooccurrence NnearP or VnearP Volk (2000) assumes that the higher value of Cooc(VnearP) and Cooc(NnearP) decides the attachment. According to this assumption we say that if Cooc(VnearP) is the higher value, the PP attaches to the verb. If Cooc(NnearP) is the higher value, the PP attaches to the noun. When only Cooc(NnearP) was available (because the phrase did not contain a verb), the decision for noun or verb attachment was based on comparison of Cooc(NnearP) with a threshold of 0.5 (cooccurrence strength values are between 0.00 and 1.00). This is the threshold used by Volk (2000).

For the 1004 phrases derived from the CGN we computed the accuracy (the percentage of correct attachment decisions), and precision, recall, and F_β-score[4] with $\beta = 1$ (van Rijsbergen, 1979), for both noun and verb attachment. The respective values are given in Table 1. A baseline was computed, which gives the performance measures when noun attachment was predicted for all 1004 phrases.

Method II: cooccurrence NnearP Alternatively, we can base the decision between noun and verb attachment on Cooc(NnearP) only, comparing the cooccurrence strength value to a threshold. The cooccurrence strength values we found according to this method range from very high to very low (1.00 - 0.00) and differ significantly for noun and verb attachment (t=-11.65, p<0.001, df=1002).

By computing the performance measures for several thresholds, using 10-fold cross validation, we determined that the optimal cooccurrence threshold should be 0.36 for optimization on noun attachment. Cooccurrence strength values higher than the threshold predict that the PP is attached to the noun. The performance measures obtained with this method are also given in Table 1.

Method III: cooccurrence N P To simplify Method II further, we use Cooc(N P) instead of Cooc(NnearP) to decide between noun and verb attachment, comparing the cooccurrence strength value to a threshold. The cooccurrence strength values we found according to this approach range from very high to very low (0.99 - 0.00) and differ significantly for noun and verb attachment (t=-12.43, p<0.001, df=1002).

By computing the performance measures for several thresholds, using 10-fold cross validation, we determined that the optimal cooccurrence threshold should be 0.07. The performance measures obtained with this method are also given in Table 1.

Preferred method Table 1 shows that Method III has the best accuracy on PP attachment. Although it is not the best in all respects, we prefer this method, because it uses cooccurrence strength values for adjacent nouns and prepositions in the order in which they appear in the text (see §3.2), this in analogy with the fact that order is meaningful in PP attachment.

Thus, we added the Cooc(N P) feature as the eleventh feature to our data sets for both corpora.

[4] $F_\beta = \frac{(\beta^2+1)\cdot\text{precision}\cdot\text{recall}}{\beta^2\cdot\text{precision}+\text{recall}}$

Table 2: *Performance measures on PP attachment in the CGN material by* RIPPER *and* IB1.

		NOUN attachment			VERB attachment		
	accuracy	precision	recall	$F_{\beta=1}$	precision	recall	$F_{\beta=1}$
RIPPER (- bigrams)	75	83	75	78	66	78	71
RIPPER (+ bigrams)	72	78	74	76	64	70	67
IB1 (- bigrams)	78	81	83	82	73	69	71
IB1 (+ bigrams)	75	79	81	80	69	67	68
baseline	60	60	100	75	-	0	-

4 Machine learning experiments

We choose to use two machine learning algorithms in our study: rule induction as implemented in RIPPER (Cohen, 1995) (version 1, release 2.4) and memory-based learning IB1 (Aha et al., 1991; Daelemans et al., 1999), as implemented in the TiMBL software package (Daelemans et al., 2002). Rule induction is an instance of "eager" learning, where effort is invested in searching for a minimal-description-length rule set that covers the classifications in the training data. The rule set can then be used for classifying new instances of the same task. Memory-based learning, in contrast, is "lazy"; learning is merely the storage of learning examples in memory, while the effort is deferred to the classification of new material, which in IB1 essentially follows the k-nearest neighbor classification rule (Cover and Hart, 1967) of searching for nearest neighbors in memory, and extrapolating their (majority) class to the new instance.

A central issue in the application of machine learning is the setting of algorithmic parameters; both RIPPER and IB1 feature several parameters of which the values can seriously affect the bias and result of learning. Also, which parameters are optimal interacts with which features are selected and how much data is available. Few reliable rules of thumb are available for setting parameters. To estimate appropriate settings, a big search space needs to be sought through in some way, after which one can only hope that the estimated best parameter setting is also good for the test material – it might be overfitted on the training material.

Fortunately, we were able to do a semi-exhaustive search (testing a selection of sensible numeric values where in principle there is an infinite number of settings), since the CGN data set is small (1004 instances). For IB1, we varied the following parameters systematically in all combinations:

- the k in the k-nearest neighbor classification rule: 1, 3, 5, 7, 9, 11, 13, 15, 25, and 45
- the type of feature weighting: none, gain ratio, information gain, chi-squared, shared variance
- the similarity metric: overlap, or MVDM with back-off to overlap at levels 1 (no backoff), 2, and 10
- the type of distance weighting: none, inverse distance, inverse linear distance, and exponential decay with $\alpha = 1.0$ and $\alpha = 2.0$

For RIPPER we varied the following parameters:

- the minimal number of instances to be covered by rules: 1, 2, 5, 10, 25, 50
- the class order for which rules are induced: increasing and decreasing frequency
- allowing negation in nominal tests or not
- the number of rule set optimization steps: 0, 1, 2

We performed the full matrix of all combinations of these parameters for both algorithms in a nested 10-fold cross-validation experiment. First, the original data set was split in ten partitions of 90% training material and 10% test material. Second, nested 10-fold cross-validation experiments were performed on each 90% data set, splitting it again ten times. To each of these 10 × 10 experiments all parameter variants were applied. Per main fold, a nested cross-validation average performance was computed; the setting with the average highest F-score on noun attachment is then applied to the full 90% training set, and tested on the 10% test set. As a systematic extra variant, we performed both the RIPPER and IB1 experiments with and without the six bigram features (mentioned in §3.1).

Table 3: *Performance on PP attachment in newspaper and e-mail material by* RIPPER *and* IB1.

		Noun attachment			Verb attachment		
	accuracy	precision	recall	$F_{\beta=1}$	precision	recall	$F_{\beta=1}$
RIPPER (-/+ bigrams)	74	80	74	77	67	74	71
IB1 (- bigrams)	71	72	82	77	70	56	62
IB1 (+ bigrams)	70	72	80	76	67	56	61
baseline	58	58	100	73	-	0	-

5 Results

Internal results: Spoken Dutch Corpus data Table 2 lists the performance measures produced by RIPPER and IB1 on the CGN data. For both algorithms it proved a disadvantage to have the bigram features; both attain higher F-scores on noun attachment without them. IB1 produces the highest F-score, 82, which is significantly higher than the F-score of RIPPER without bigrams, 78 (t=2.78, p<0.05, df=19).

For RIPPER, the best overall cross-validated parameter setting is to allow a minimum of ten cases to be covered by a rule, induce rules on the most frequent class first (noun attachment), allow negation (which is, however, not used effectively), and run one optimization round. The most common best rule set (also when including bigram features) is the following:

1. if P = *van* then NOUN
2. if cooc(N P) > 0.0812 then NOUN
3. if P = *voor* then NOUN
4. if there is no verb then NOUN
5. else VERB

This small number of rules test on the presence of the two prepositions *van* (from, of) and *voor* (for, before) which often co-occur with noun attachment (on the whole data set, 351 out of 406 occurrences of the two), a high value of Cooc(N P) similar to the threshold reported earlier (0.07), and the absence of a verb (which occurs in 27 instances).

The best overall cross-validated setting for IB1 was no feature weighting, $k = 11$, and exponential decay distance weighting with $\alpha = 2$. It has been argued in the literature that high k and distance weighting is a sensible combination (Zavrel et al., 1997). More surprisingly, no feature weighting means that every feature is regarded equally important.

External results: newspaper and e-mail data We evaluated the results of applying the overall best settings on the 157 sentence external newspaper and e-mail material. Performances are given in Table 3. These results roughly correspond with the previous results; IB1 has lower precision but higher recall than RIPPER on noun attachment. RIPPER performed the same with and without bigram features, since its rules do not test on them. Overall, these results suggest that the learned models have a reasonably stable performance on different data.

6 Contribution to phrase boundary allocation

In a third experiment we measured the added value of having PP attachment information available in a straightforward existing prosodic phrasing algorithm for Dutch (van Herwijnen and Terken, 2001b). This phrasing algorithm uses syntactic information and sentence length for the allocation of prosodic phrase boundaries. For a subset (44 phrases) of the held-out corpus, we compared the allocation of boundaries according to the phrasing algorithm and according to the same algorithm complemented with PP attachment information, to a consensus transcription of ten phonetic experts (van Herwijnen and Terken, 2001a). This consensus transcription was not available for all 157 phrases of the newspaper and e-mail data.

Table 4 shows the performance measures for this comparison, indicating that the improvement from PP attachment information is largely in precision. Indeed, blocking certain incorrect placements of phrase boundaries improves the precision on boundary placement. IB1 attains the best improvement of six points in precision. Although it incorrectly prevents five *intended* phrase bound-

Table 4: *Performance on phrasing complemented with PP attachment information from* RIPPER *and* IB1 *with and without bigram features.*

phrasing algorithm	accuracy	precision	recall	$F_{\beta=1}$
phrasing	91	65	81	72
phrasing + RIPPER (-/+ bigrams)	92	70	80	74
phrasing + IB1 (- bigrams)	92	70	79	74
phrasing + IB1 (+ bigrams)	92	71	79	75
phrasing + gold standard	93	72	81	77

aries (when compared to the manual classification mentioned in §2), it does in fact correctly prevent *unintended* boundaries in twelve other cases. Some examples of the latter are:

1. ... afschaffing | van het laatste recht ...
2. ... het grootste deel | van Nederland ...
3. ... de straatlantaarns | langs de provinciale weg ...

1. ... abolition | of the final right ...
2. ... the biggest part | of the Netherlands ...
3. ... the street lights | along the provincial road ...

Table 4 also shows the performance measures for the phrasing algorithm complemented with the "gold standard". These results indicate the maximal attainable improvement of the phrasing algorithm using correct PP attachment information. The results obtained with IB1 come close to this maximal attainable improvement, particularly in terms of precision.

7 Discussion

We have presented experiments on isolated learning of PP attachment in Dutch, and on using predicted PP attachment information for filtering out incorrect placements of prosodic boundaries. First, PP attachment was learned by the best optimized machine learner, IB1 at an accuracy of 78, an F-score of 82 on noun attachment, and 71 on verb attachment. The learners were optimized (via nested cross-validation experiments and semi-exhaustive parameter selection) on noun attachment, since that type of attachment typically prevents a prosodic boundary. In general, incorrect boundaries are considered more problematic to the listener than omitted boundaries. We show that small improvements are made in the precision of boundary allocation; a high precision means few incorrect boundaries.

Comparing the eager learner RIPPER with the lazy learner IB1, we saw that RIPPER typically induces a very small number of safe rules, leading to reasonable precision but relatively low recall. The bias of IB1 to base classifications on all training examples available, no matter how low-frequent or exceptional, resulted in a markedly higher recall of up to 82 on noun attachment, indicating that there is more reliable information in local matching on lexical features and the cooccurrence feature than RIPPER estimates. However, with a larger training corpus, we might not have found these differences in performance between IB1 and RIPPER.

In engineering our feature set we combined disjoint ideas on using both lexical (unigram and bigram) features and cooccurrence strength values. The lexical features were sparse, since they only came from the 1004-instance training corpus, while the cooccurrence feature was very robust and "unsupervised", based on the very large WWW. Within the set of lexical features, the bigram features were sparser than the unigram features, and neither of the algorithms benefited from the bigram features. Thus, given the current data set, all necessary information was available in the four unigram features in combination with the cooccurrence feature. Only the combination of the five yielded the best performance – individually the features do carry information, but always less than the combination. When running nested cross-validation experiments with IB1 on the four unigram features, F-scores are lower than the optimal 82: 77 (N1), 75 (P), 72 (V), 74 (N2), and 75 Cooc(N P). These results suggest that it is essential for this experiment to employ features that (1) are preferably robust counter to sparse, and (2) each add unique information, either on lexical identity or on cooccurrence strength.

Although the addition of more sparse and redundant features (bigrams) turned out to be ineffective at the current data size, there is no reason to expect that they will not facilitate performance on larger data sets to be developed on the near feature. Besides, it would be interesting to investigate ways of embedding our approach for predicting PP attachment within other, more general parsing algorithms.

References

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

Altavista. 2002. Advanced search cheat sheet. http://www.altavista.com/. Page visited Sept. 2002.

J. Bear and P. Price. 1990. Prosody, syntax and parsing. In *Proc. of the Association of Computational Linguistics conference*, pages 17–22.

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

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of NAACL'00*, pages 132–139.

K.W. Church and P. Hanks. 1991. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16:22–29.

W. W. Cohen. 1995. Fast effective rule induction. In *Proc. of the Twelfth International Conference on Machine Learning*, Lake Tahoe, California.

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

M.J. Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proc. of the 34th Annual Meeting of the Association for C omputational Linguistics.*

T. M. Cover and P. E. Hart. 1967. Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27.

W. Daelemans, A. van den Bosch, and J. Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning, Special issue on Natural Language Learning*, 34:11–41.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2002. TiMBL: Tilburg Memory Based Learner, version 4.3, reference manual. Technical Report ILK-0210, ILK, Tilburg University.

T. Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

A. Franz. 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*, pages 188–202. Springer-Verlag, New York.

O.M. van Herwijnen and J.M.B. Terken. 2001a. Do speakers realize the prosodic structure they say they do? In *Proc. of Eurospeech 2001 Scandinavia*, volume 2, pages 959–962.

O.M. van Herwijnen and J.M.B. Terken. 2001b. Evaluation of PROS-3 for the assignment of prosodic structure, compared to assignment by human experts. In *Proc. of Eurospeech 2001 Scandinavia*, volume 1, pages 529–532.

D. Hindle and M. Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

E. Marsi, P.-A. Coppen, C. Gussenhoven, and T. Rietveld. 1997. Prosodic and intonational domains in speech synthesis. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg, editors, *Progress in Speech Synthesis*, pages 477–493. Springer-Verlag, New York.

P. C. Paardekooper. 1977. *ABN, Beknopte ABN-syntaksis.* Eindhoven, 5th edition.

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

A. Ratnaparkhi. 1997. A linear observed time statistical parser based on maximum entropy models. In *Proc. of the Second Conference on Empirical Methods in Natural Language Processing, EMNLP-2, Providence, Rhode Island*, pages 1–10.

C.J. van Rijsbergen. 1979. *Information Retrieval.* Buttersworth, London, 2nd edition.

A.A. Sanderman and R. Collier. 1997. Prosodic phrasing and comprehension. *Language and Speech*, 40(4):391–409.

A.A. Sanderman. 1996. *Prosodic phrasing: production, perception, acceptability and comprehension.* Ph.D. thesis, Eindhoven University of Technology.

E.O. Selkirk. 1984. *Phonology and Syntax: the Relation between Sound and Structure.* Cambridge, MA: MIT Press.

T. van der Wouden, H. Hoekstra, M. Moortgat, B. Renmans, and I. Schuurmans. 2002. Syntactic analysis in the Spoken Dutch Corpus. In *Proc. of the Third International Conference on Language Resources and Evaluation, LREC-2002*, pages 768–773.

M. Volk. 2000. Scaling up. Using the WWW to resolve PP attachment ambiguities. In *Proc. of KONVENS-2000*, pages 151–156. Sprachkommunikation, Ilmenau, VDE Verlag.

J. Zavrel, W. Daelemans, and J. Veenstra. 1997. Resolving PP attachment ambiguities with memory-based learning. In Mark Elison, editor, *Proceedings of Conference on Computational Natural Language Learning*, pages 136–144.