

Grammar induction and PP attachment disambiguation

Jorik Jonker

March 19, 2007

Abstract

The task of parsing is one of the oldest tasks in the field of computational linguistics, facing two major fundamental problems: attachment disambiguation and grammar construction. The former problem commonly manifests itself as the “PP attachment disambiguation problem”, where the decision whether the prepositional phrase attaches to the verb or noun is ambiguous. The latter problem can be solved by inducing grammar from a corpus, “grammar induction”. While the literature is rich on both subjects, this paper gives an survey on the literature on both tasks, where PP-attachment disambiguation.

1 Introduction

In the world of computational linguistics, parsing is often required for many tasks. In order to be able to parse a sentence, a defined set of rules, called a “grammar” is needed. For simple texts, a simple grammar often suffices, but as the complexity of texts increases, the size of the grammar increases too.

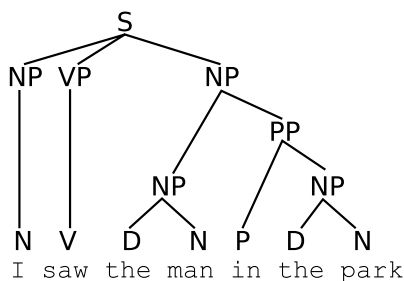


Figure 1: Example parse tree

Figure 1 shows an example of a simple parse tree.

The rules in table 1 are the minimum set of grammar rules required to be able to parse this sentence. It shows that for a simple sentence like this already six rules are needed. It is evident that more complex sentences require much more grammar rules to be able to be parsed.

S	\rightarrow	$NP VP$
NP	\rightarrow	N
VP	\rightarrow	V
NP	\rightarrow	$NP PP$
NP	\rightarrow	DN
PP	\rightarrow	$P NP$

Table 1: Example grammar rules

The construction of grammars in for parsing documents is a very time-consuming process. Databases with annotated, parsed sentences (called *treebanks*) are thus of great value to computational linguists. If we are able to somehow automatically enhance or even create such treebanks, that would be of great value. Grammar induction is the task of extracting linguistic structure (i.e. grammar) from a (large) text, ideally reducing the intensive task of creating it by hand. In an *supervised* setting, a grammar is induced from an annotated document where structure is gathered from a (large) set of examples, the *corpus*. In this context, the corpus is regarded as a collection of sentences, which have been manually parsed.

When *unsupervised* learning is applied, however, the algorithm is to be applied on unannotated text. Ideally, grammar induction should yield a correct grammar from unannotated text, but it is evident that this is a task of greater difficulty compared with the supervised version.

There is, however, still one problem which is usually not covered with grammars per se: the disambiguation of certain ambiguous attachment,

particularity that of prepositional phrases. Consider for instance the following two sentences: I ate a pizza with anchovies and I ate a pizza with a fork. Both sentences contain the same “ingredients” (noun phrase, verb phrase, noun phrase and prepositional phrase) but are parsed in similar ways (see figure 3). Formal grammars do not really address this problem, since the attachment is mainly determined by lexical features, requiring a different approach. This problem can, like grammar induction, theoretically be solved both supervised and unsupervised. In this paper an in-depth overview is given of the several approaches from the literature, towards this task.

Both problems have a supervised and unsupervised approach, where the former requires a corpus containing parsed sentences, called a *treebank*. A popular (English) treebank is the Penn Treebank (see Marcus et al., 1993), containing parse trees of the Wall Street Journal, the Brown Corpus, Switchboard and ATIS. The corpus consists of 4.5 million words of which roughly two-thirds is *bracketed*¹.

This paper is organised as follows: the first section deals with the current literature on the subject of grammar induction, followed by the second section, covering PP attachment disambiguation. The last section wraps both subjects up, giving the conclusions on the matter.

2 Grammar induction

Grammar induction can be formulated as the task of discovering common structures in utterances which are generated by the same process. The grammar used in natural languages can be represented as a *context-free grammar*, especially with so called *stochastic context-free grammars*. Ambiguous rules, having multiple possibilities on the right hand side are paired with probabilities. The former are mostly hand-crafted and need to be of reasonable size in order to be used for parsing most documents. The latter representation consists of production rules, paired with probabilities, which can be automatically trained using grammar induction.

A lot of work (Baker, 1979; Lari and Young, 1990, 1991; Pereira and Schabes, 1992) on gram-

mar induction is based on the *inside-outside algorithm*, introduced by Baker (1979) as a generalisation of the parameter estimation methods for hidden Markov models to stochastic context-free grammars (Booth, 1969). This algorithm is a variant of the expectation-maximization algorithm, assuming that a “good” grammar is one that makes sentences in the training corpus likely to occur. The algorithm takes as input a stochastic context-free grammar (initialized with equal probabilities) and a training corpus and it iteratively reestimates probabilities in order to maximize the probability that the grammar would produce the corpus. This algorithm can both be used to infer probabilities and to infer the grammar itself as well (see T. Fujisaki et al., 1989).

The inside-outside algorithm iteratively uses the current rule probabilities to estimate the expected frequencies of certain types in the corpus and then estimate the new rule probabilities from those expected frequencies. Those expected and re-computed frequencies are referred to as *inside* and *outside* probabilities.

The application of the original algorithm remained inconclusive (Lari and Young, 1990, 1991), due to its impractical computational complexity and due to its poor convergence properties on larger grammars. A key aspect to (variants of) the expectation-maximization algorithm is convergence, since this is often the only “metric” to decide whether to stop iterating or not. (Pereira and Schabes, 1992) has proposed an extension to the algorithm, which modifies the algorithm to operate on partially bracketed corpora. The author reports an improvement in both convergence behaviour and complexity, as well as a huge improvement of the accuracy of the improved grammar (measured in correct placed brackets).

The work of Hwa (1999) compares the direct induction and adaptation of grammars under different training conditions. In order to induce grammars both from scratch and as an adaptation, a variant of the Inside-Outside re-estimation algorithm (see Pereira and Schabes, 1992; Baker, 1979), producing so-called probabilistic lexicalized tree insertion grammars (PLTIG) (see Schabes and Waters, 1993). Tree insertion grammars are collections of tree fragments, which can substitute nodes in other trees. LTIG’s are known (see Pereira and Schabes, 1992; Hwa, 1999) to have the same ex-

¹Bracketing is a representation of a parse tree

pressive power as context free grammars, and lexicalized grammars have the advantage that ambiguities (such as PP attachments) can be clarified by grammar rules.

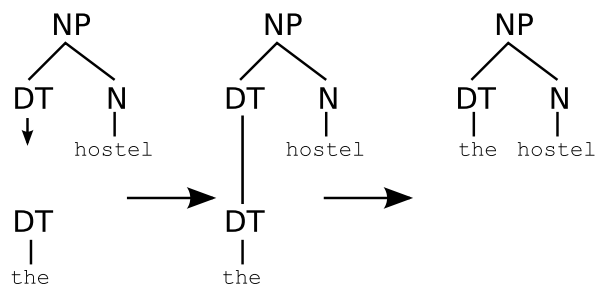


Figure 2: Example LTIG

When adapting, a grammar is induced on a labelled corpus, and then re-estimated using the new labelling. Both methods were evaluated using the Wall Street Journal (WSJ) and Air Travel Information System (ATIS) corpora. The author reports that pre-training a grammar on one corpus, re-estimating on another results in a higher accuracy than directly induced grammars.

Chen (1995) describes a corpus-based induction algorithm for probabilistic context-free grammars for medium-sized domains, deploying the induction problem as a heuristic search. This search consists of finding the grammar with the largest *a posteriori* probability given the training data, favouring smaller grammars over large ones. The goal of favouring small grammars (conform Occam's Razor²) was implemented by giving a grammar an a-priori probability of $2^{-l(G)}$, where $l(G)$ is proportional to the size of the grammar. The (greedy) search is conducted by starting with a small, trivial grammar, modifying the grammar, resulting in a grammar with a higher probability. The search is initialised with a single rule with can generate any string. The parameters (probabilities) of the resulting grammar are set based on the Viterbi parse of the training data. The author reports that the resulting grammars lack a certain degree of expressibility, which is addressed by using the Inside-Outside algorithm as a post-pass.

Klein and Manning have done a lot of work in the field of inducing syntactic structure from corpora

²"All things being equal, the simplest solution tends to be the best one."

(Klein and Manning, 2001a,b, 2002, 2004). In Klein and Manning (2001b), it is argued that potential constituents in unsupervised grammar induction should be judged by substitutability instead on the likelihood in the data. The authors introduce the notion of a *signature*: a set of contexts in which a potential constituent can occur. The entropy of this signature is employed to distinguish between constituents and non-constituents.

In Klein and Manning (2001a) the authors state that classical (EM) parameter search require local rule-merge criteria in order to produce coherent grammars. Whenever such an approaches seems to produce decent results, it is because of good local heuristics. This highlight an interesting phenomenon that long constituents often have short, common equivalents which appear in similar contexts. This knowledge is incorporated in their EM-approach, which operates on two representations of the probability distributions of the context of the potential constituent.

2.1 Chunking

Parsing a text often requires a parser, which takes as input a grammar, constructing which is a very labour-intensive task. Several approaches to automatically parse a text have been made, amongst which "chunking", introduced by Abney (1991). Chunking is a technique in which base phrases (such as noun phrases) are identified by a parser, which is basically placing brackets in the sentence to be "parsed".

The work of Veenstra (1998) suggest memory-based learning (MBL) as a technique to chunk noun phrases. During training, the classifier is presented with a labeled dataset, which are added to memory. During testing, for each unseen pattern the distance is calculated to all cases in memory, predicting the class with the category of the nearest case(s). The distance metric used is straightforward similarity of the used features. Each features is weighted by its information gain (see Daelemans and van den Bosch, 1992). The authors state that without optimisation, this approach has an asymptotic complexity of $O(NF)$ (where N is the number of items in memory and F the amount of features), which is quite costly. The authors overcome this problem by storing the examples in a structure called IGTREE, a structure combining the compres-

sion of case bases and the retrieving of cases. Using a memory based transformation POS tagger, the authors report a chunking accuracy of 96.8%.

Recent work of Stegeman (2006) suggest the usage of support vector machines (see Boser et al., 1992) towards both chunking and part-of-speech tagging. The author had trained and evaluated the chunker on the Spoken Dutch Corpus (see Oostdijk et al., 2002), using the output of his POS tagger as input for the chunker. The chunker is constructed in such a way that it outputs IOB tags describing whether a word is Inside, Outside or at the Beginning of a new chunk. To classify the IOB-tag of a word, Stegeman uses the POS tag, the lexeme as well as the chunk tags of the seven preceding words. Bi- and trigrams of the mentioned features are added as extended features. Finally, the author adds non-local features such as the distance to this tag, bigrams of succeeding words, the number of preceding chunks and the length of the current chunk as “non-local features”. Stegeman reports an accuracy of 86.37%, using a 3rd degree polynomial kernel.

2.2 Data Oriented Parsing

Data Oriented Parsing, or DOP, conceived by Scha (1990) is a highly accurate method to parse data without a grammar. The original aim of DOP was to develop a model reflecting psycholinguistical insight instead of the traditional rewrite rules when using grammars. DOP is centered around what is called the treebank: a corpus of parsed sentences which is used to induce new parses from. Since DOP does not generate a grammar, it formally does not do grammar induction. However, since it delivers parses trained on a treebank, it’s domain is certainly related. An important drawback of this application is the fact that the computational cost of DOP is ridiculously high: DOP1, as implemented by Bod (1995), was reported to have an average speed of 3.5 hours per sentence (see de Pauw, 2000).

3 PP-attachment

A lot of work is done in the field of PP-attachment problems, both supervised and unsupervised. Supervised methods deploy an annotated corpus with “solved” attachments, which is fed to some

kind of machine learning technique. Unsupervised methods may use an unannotated corpus to disambiguate the attachments, using information which was not manually annotated. The corpus may be preprocessed in some way, but the attachments are not disambiguated yet. In the next sections a survey of both supervised and unsupervised methods to disambiguate PP-attachments is given. A common way to abstract from the PP-attachment problem is to decide whether the preposition P in the tuple (V, N_1, P, N_2) attaches to V or N , resulting in an attachment tuple (V, P, N_2) or (N_1, P, N_2) . Altmann and Steedman (1998) were the first to report on this issue in the literature, showing that discourse text is often needed to disambiguate the attachment. More recent work (Hindle and Rooth, 1993; Brill and Resnik, 1994; Collins and Brooks, 1995) shows that lexical features often suffice in order to disambiguate the attachment site.

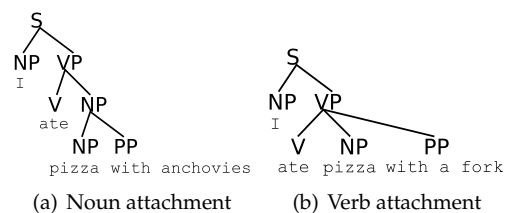


Figure 3: Attachment sites

3.1 Supervised

One of the first corpus-based approaches towards this problem is that of Hindle and Rooth (1993). In this study the lexical association between verbs and nouns with prepositions is measured and used to decide between attachments of unseen samples. Training data was obtained by extracting all (V, N_1, P, N_2) tuples from a large manually annotated corpus.

Ratnaparkhi et al. (1994) propose a maximum entropy (ME) model for solving PP-attachment problems, using two types of features: “word” features, which test the presence of certain n-grams and “class” features, which query (binary) class membership of head words. To construct the binary classes, words are clustered according to their frequency. To increase generalizability, an incremental feature search is performed, in which features

are ranked by the estimate of their contribution to the log-likelihood of the training set. The authors report an accuracy of 81.6% on the Penn Treebank WSJ set.

Kawahara and Kurohashi (2005) use a gigantic volume of truly unambiguous examples to acquire precise lexical preferences to disambiguate PP-attachments, using co-occurrence statistics of the pairs (V, P) and (N, P) . The corpus was obtained by crawling 200M web pages which were extracted, tagged and chunked, resulting in 1300M chunked sentences. To extract truly unambiguous examples, using simple heuristics, such as the fact that a prepositional phrase cannot attach to pronouns. The authors deploy a support vector machine, using the (preprocessed) lexeme of the quadruple, the part-of-speech information of the quadruple and the lexical preference statistics calculated as features. The authors report an accuracy of 87.25% on the WSJ Penn Treebank set.

3.2 Unsupervised

Pantel and Lin (2000) suggest an unsupervised iterative method, extracting training data from an automatically parsed corpus. Given a dataset containing ambiguous attachment sites, they count for every ambiguous site with k possible attachments the scores for the tuples (V, P, N_2) and (N_1, P, N_2) , incrementing those scores by $\frac{1}{k}$. After each iteration, the scores are “normalized” according to a special formula. Furthermore, the authors use contextually similar words from the collocation database to approximate attachment scores using an other algorithm. Pantel and Lin (2000) report an accuracy of 84.31% on the Penn Treebank WSJ dataset (Ratnaparkhi et al., 1994) containing 3097 examples, approximating the human accuracy (without using context) of 88.2% on that set.

A popular approach, which could be regarded as unsupervised learning is the usage of the WWW to estimate co-occurrence strengths (Calvo and Gelbukh, 2003; Volk, 2000, 2001; van Herwijnen et al., 2003). A co-occurrence strength is a metric to capture the (relative) frequency of two words occurring together in one sentence.

$$cooc(X, Y) = \frac{freq(X, Y)}{freq(X)} \quad (1)$$

If some noun N occurs 100 times in a corpus and this noun occurs 60 times in a sentence containing preposition P too, we say that the co-occurrence strength of N and P is 0.6. To have a robust estimation of co-occurrence strength, in order to use them in deciding between N and V attachment, a (very) large corpus is needed. The WWW is a corpus of orders of magnitude larger than any locally accessible corpus and thus is of interest for this matter.

In these methods a WWW search engine is deployed to query for documents containing two words, whose count is used to estimate the co-occurrence strength (see Volk, 2000, 2001). Initially, Altavista was used, because of its NEAR-operator, which searches for documents where the two words surrounding the operator should appear “near” each other in the retrieved documents. Calvo and Gelbukh (2003) suggests replacing Altavista by Google for this purpose, because it has indexed more documents. Although Google lacks this NEAR-operator Calvo and Gelbukh reports more accurate results than Volk (2000).

3.3 Mixed

Volk (2002) discusses a combination of both supervised and unsupervised methods as an approach. The unsupervised method involved is simply comparing co-occurrence scores of (N, P) and (V, P) pairs. It appears, however that the two attachment classes are imbalanced, which would favour verb attachment over noun attachment. The author has solved this by adding a noun factor to the score formula, which flattens out the a-priori imbalances. To increase attachment coverage, the author uses triples $((V, P, N_2)$ and $(N_1, P, N_2))$, backing off to the scores using pairs. The discussed supervised method in this article is the back-off method introduced by Collins and Brooks (1995), using the best information available, backing off to the next level whenever information is missing. The methods were combined intertwining the co-occurrence scores between the different levels in the back-off model, resulting in 10 decision levels. The author reports an accuracy of 80.98% on the (German) NEGRA test set.

Bharathi et al. (2005) propose a technique in which both supervised and unsupervised methods are used, as well as WordNet (Fellbaum, 1998,

see) to disambiguate PP-attachments. The authors calculate trigram scores from an annotated corpus. Next, using a modified version of the EM-algorithm, attachment probabilities are extracted from an unannotated corpus. The attachment decision is then based on values calculated in the previous steps, using WordNet to handle unseen words. The authors report an accuracy of 89.35% on the WSJ treebank. Moreover, the authors suggest a method to deal with multiple PP-attachment problems (e.g. “will be an office in the east of London”; $(V, N_a, P_1, N_1, P_2, N_2)$), analogous to their approach to single attachment disambiguation approaches.

Zhao and Lin (2004) present a nearest-neighbour algorithm similar to Zavrel et al. (1996). The authors employ what is called *distributional similarity* of words to increase accuracy in attachment decision. Distributional similarity is the phenomenon that words in same contexts have similar meanings (Harris, 1968). The authors show that the words *test* and *exam* are similar, because both of them can be objects of verbs such as *administer*, *cancel*, *conduct*. . . and both of them can be modified by adjectives such as *academic*, *diagnostic*, *difficult*. . . . The authors report an accuracy on correct attachment decision of 86.5%, which is significantly³ higher than results reported in Collins and Brooks (1995); Zavrel et al. (1996).

3.4 Dutch

Very few attempts have been made to investigate the problem of ambiguous PP attachments for Dutch. Volk has addressed the problem several times on a quite similar language, German (Volk, 2000, 2001, 2002, 2006). Vandeghinste (2002); van Halteren et al. (2005) appear to have written abstracts on the subject with respect to Dutch, however, results of the proposed experiments were not to be found.

van Herwijnen et al. (2003) published a paper on Dutch, focussing on the application of prosodic phrasing. At present, there are no parsers available for Dutch that disambiguate PP attachment, so this study is a classic “isolated” case. The authors compare two (machine) learning algorithms: a rule induction system (RIPPER, see Cohen (1995)) and a memory-based system (IB1, see Daelemans

³with $\geq 98\%$ confidence

et al. (1999)). The used features were the four heads ($N1$, PP , V and $N2$) and *all* combinations of two heads. Additionally, the authors add cooccurrence strengths from the WWW using a technique similar to that of Volk (2000) in which the WWW-queries were slightly reformulated. Of both algorithms the parameters were tuned using a semi-exhaustive search because of the absence of reliable rules of thumb to adjust them manually. The authors report that IB1 outperforms RIPPER, with an F-score of 82.

3.5 Evaluation

Volk (2006) argues that the noun attachment rate (NAR), which is often used to describe the “difficulty” of the disambiguation process is probably not the best measure to baseline this difficulty. In the literature (Hindle and Rooth, 1991; Ratnaparkhi et al., 1994; Volk, 2001) the performance of PP-disambiguation is often measured against the percentage of ambiguous attachments which is to be attached to a noun. When for instance 80 % of the PP’s attaches to a noun, the baseline is set at 80 %. Volk points out that closer investigation shows that those NAR’s are not always properly measured in the literature. The work of Ratnaparkhi et al. (1994) and Hindle and Rooth (1993) show a much lower NAR than Volk (2006), because the authors of the former works had an incorrect idea of what to count as noun attachment and what not.

Furthermore, Volk reports that the difference in NAR between English and the languages German and Swedish lies in the fact that the word *of* is the most common preposition and is a 99% sure noun-attachment. This fact creates a whole different problem if we disregard ambiguous phrases containing that preposition. Both German and Swedish do not have such an *augmenting* preposition for that matter.

3.6 Nature

This section so far has focussed on the attachment site of ambiguous prepositional phrase attachments, whereas the *nature* of the attachment is of interest too: PP attachments can be either an *argument* or an *adjunct*. Consider the following examples: She baked a cake for her mother and

She baked a cake for an hour. In the first example the PP for her mother is an argument for the verb bake, whereas the for an hour is an adjunct PP. The same goes for noun attachments, which can be an argument or adjunct, respectively exemplified: She is a student of physics and She is a student from Phoenix. Merlo and Ferrer (2006) argue that besides attachment the nature of the attachment is important, since the distinction between arguments and adjuncts is important in identifying the semantic “kernel” of a sentence. Merlo and Ferrer transform the PP attachment task from two-way to a four-way classification. To distinguish between arguments and adjuncts, they model several linguistics properties, such as the dependence of arguments on lexical heads, the optionality of arguments and the fact that adjuncts can be iterated. Those modeled properties, alongside with lexical classes formed the feature set used for supervised learning. The authors apply a supervised learning method on manually labeled data, extracted from the Penn Treebank, which was preprocessed since it lacked explicit argument/adjunct annotation. Merlo and Ferrer report reasonable good performance for verb adjuncts, noun arguments and noun adjuncts independent on the learning algorithm, whereas they only yielded decent results on verb adjuncts using large margin classifiers.

4 Conclusion and further work

The induction of grammar is often a technique in which existing grammars (although those can be extremely simple) are enhanced by a annotated corpus. This enhancement can be in several ways: labeling grammar rules with probabilities – constructing stochastic context free grammars –, deriving new rules from large corpora or a combination of both. In any case, variants of the expectation maximization algorithm are employed.

For some tasks, a much shallower parse suffices, labeling only constituents by placing brackets. This technique, called chunking, is mainly powered by machine learning tasks, trained on a bracketed corpus. The state-of-the-art use a memory-based-learner, with unfortunate complexity, where recent work has shown (Stegeman, 2006) that modern, efficient techniques are able to yield a decent

performance as well.

Both techniques do not deal well with ambiguous attachments such as PP-attachments, except the technique of Hwa (1999), because the methods operate on delexicalized text. To deal with PP attachment ambiguities, either (probabilistic) lexicalized grammar insertion trees or specific disambiguation techniques have to be used, because PP attachment site is determined by lexical features. Almost every “isolated” PP disambiguation technique attempts to decide on co-occurrence strengths, both supervised and unsupervised. In the latter form, the *www* is often consulted, being a very large, easy accessible corpus. Since most (if not all) of the PP attachment disambiguation work is done on isolated sentences, it would be interesting to see if the context of a sentence influences the attachment.

Much of the literature on both subjects deals with English, implicitly assuming generalisation to other languages as well. Some effort has been made to evaluate existing PP-attachment disambiguation techniques on Dutch, Swedish and German, but that is just the tip of the iceberg. Moreover, it would be interesting to see if the suggested techniques apply to other attachment disambiguities as well, such as *yellow gold ring*, where *yellow* can attach both to *gold* as *ring*.

References

- S. P. Abney. Parsing by chunks. In R. C. Berwick, S. P. Abney, and C. Tenny, editors, *Principle-Based Parsing: Computation and Psycholinguistics*. Kluwer Academic Publishers, Dordrecht, 1991.
- G. Altmann and M. Steedman. Interaction with context during human sentence processing. *Cognition*, 30:191–238, 1998.
- J. Baker. Trainable grammars for speech recognition. In J. J. Wolf and D. H. Klatt, editors, *Speech communication papers presented at the 97th meeting of the Acoustical Society of America*, pages 547–550, Cambridge, MA, June 1979. MIT.
- A. Bharathi, U. Rohini, P. Vishnu, S. M. Bendre, and R. Sangal. A hybrid approach to single and multiple pp attachment using wordnet. *Lecture*

- Notes in Computer Science*, 3651:211–222, September 2005. doi: 10.1007/11562214.19.
- R. Bod. *Linguistics With Enriching Statistics: Performance Models Of Natural Language*. PhD thesis, Universiteit van Amsterdam, 13 Sept. 1995.
- T. Booth. Probabilistic representation of formal languages. In *Tenth Annual IEEE Symposium on Switching and Automata Theory*, October 1969.
- B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, New York, NY, USA, 1992. ACM Press. ISBN 0-89791-497-X. doi: 10.1145/130385.130401.
- E. Brill and P. Resnik. A rule-based approach to prepositional phrase attachment disambiguities. In *Proceedings of COLING-94*, pages 1198–1204, Kyoto, Japan, 1994.
- H. Calvo and A. F. Gelbukh. Improving Prepositional Phrase Attachment Disambiguation Using the Web as Corpus. In *Proceedings of CIARP 2003*, pages 604–610, 2003.
- S. F. Chen. Bayesian grammar induction for language modeling. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 228–235, Morristown, NJ, USA, 1995. Association for Computational Linguistics.
- W. Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, Tahoe City, CA, 1995.
- M. Collins and J. Brooks. Prepositional phrase attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 27–38, Cambridge, Massachusetts, 1995.
- W. Daelemans and A. van den Bosch. Generalization performance of backpropagation learning on a syllabification task. In M. Drossaers and A. Nijholt, editors, *Connectionism and Natural Language Processing. Proceedings of the Third Twente Workshop on Language Technology*, pages 27–38, 1992. URL <http://www.cnts.ua.ac.be/Publications/1992/DV92>.
- W. Daelemans, A. V. D. Bosch, and J. Zavrel. Forgetting exceptions is harmful in language learning. *Mach. Learn.*, 34(1-3):11–41, 1999. ISSN 0885-6125.
- G. de Pauw. Aspects of Pattern-matching in Data-Oriented Parsing. In *Proceedings of the 18th conference on Computational linguistics*, pages 236–242, Morristown, NJ, USA, 2000. Association for Computational Linguistics. ISBN 1-55860-717-X.
- C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.
- Z. Harris. *Mathematical structures of language*. Wiley, New York, 1968.
- D. Hindle and M. Rooth. Structural Ambiguity and lexical relations. *Computational Linguistics*, 19(1): 103–120, 1993.
- D. Hindle and M. Rooth. Structural ambiguity and lexical relations. In *Meeting of the Association for Computational Linguistics*, pages 229–236, 1991.
- R. Hwa. Supervised grammar induction using training data with limited constituent information. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 73–79, Morristown, NJ, USA, 1999. Association for Computational Linguistics. ISBN 1-55860-609-3.
- D. Kawahara and S. Kurohashi. Pp-attachment disambiguation boosted by a gigantic volume of unambiguous examples. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing*, pages 188–198, 2005. doi: 10.1007/11562214.17.
- D. Klein and C. Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the ACL*, 2004.
- D. Klein and C. D. Manning. A generative constituent-context model for improved grammar induction. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 128–135, Morristown, NJ, USA, 2001a. Association for Computational Linguistics.

- D. Klein and C. D. Manning. Distributional phrase structure induction. In *The Fifth Conference on Natural Language Learning*, 2001b.
- D. Klein and C. D. Manning. Natural language grammar induction using a constituent-context model. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.
- K. Lari and S. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.
- K. Lari and S. J. Young. Applications of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 5:237–257, 1991.
- M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19, 1993.
- P. Merlo and E. E. Ferrer. The notion of argument in prepositional phrase attachment. *Computational Linguistics*, 32(3):341–378, 2006. ISSN 0891-2017. doi: 10.1162/coli.2006.32.3.341.
- N. Oostdijk, W. Goedertier, F. van Eynde, L. Boves, J.-P. Martens, M. Moortgat, and H. Baayen. Experiences from the spoken dutch corpus project. In *LREC 2002 : Third Int. Conference on Language Resources and Evaluation*, volume 1, pages 340–347, Las Palmas de Gran Canaria, 5 2002. European Language Resources Association (ELRA).
- P. Pantel and D. Lin. An unsupervised approach to prepositional phrase attachment using contextually similar words. In K. Vijayshanker and C.-N. Huang, editors, *Proceedings of the 38th Meeting of the Association for Computational Linguistics*, pages 101–108, Hong Kong, October 2000.
- F. Pereira and Y. Schabes. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics*, pages 128–135, Morristown, NJ, USA, 1992. Association for Computational Linguistics.
- A. Ratnaparkhi, J. Reynar, and S. Roukos. A maximum entropy model for prepositional phrase attachment. In *HLT '94: Proceedings of the workshop on Human Language Technology*, pages 250–255, Morristown, NJ, USA, 1994. Association for Computational Linguistics. ISBN 1-55860-357-3.
- R. Scha. Taaltheorie en taaltechnologie: competence en performance. In Q. A. M. de Kort and G. L. J. Leerdam, editors, *Computertoepassingen in de Neerlandistiek*, LVVN-jaarboek, pages 7–22. Landelijke Vereniging van Neerlandici, Almere, 1990.
- Y. Schabes and R. Waters. Stochastic lexicalized context-free grammar. In *Proceedings of the 3rd International Workshop on Parsing Technologies*, pages 257–266, 1993.
- L. Stegeman. Part-of-speech tagging and chunk parsing of spoken dutch using support vector machines. In *Proceedings of the 4th Twente Student Conference on IT*, 2006. Bachelor Referraat.
- T. Fujisaki, F. Jelinek, J. Cocke, E. Black, and T. Nishino. A probabilistic parsing method for sentence disambiguation. In *Proceedings of the International Workshop on Parsing Technologies*, Pittsburgh, Aug. 1989.
- H. van Halteren, P. A. Coppen, B. Elffers, D. Bavcar, and M. Hulsbosch. Prepositional Phrase Attachment for Dutch: New attention for an old task. Abstract, 2005.
- O. van Herwijnen, J. Terken, A. van den Bosch, and E. Marsi. Learning pp attachment for filtering prosodic phrasing. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 139–146, Morristown, NJ, USA, 2003. Association for Computational Linguistics. ISBN 1-333-56789-0.
- V. Vandeghinste. Resolving PP Attachment Ambiguities Using the WWW. Abstract, 2002.
- J. Veenstra. Fast np chunking using memory-based learning techniques, 1998.
- M. Volk. Scaling up. Using the WWW to resolve PP attachment ambiguities. In *Proceedings of Konvens-2000*, Ilmenau, Oct. 2000.

- M. Volk. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In *Proceedings of Corpus Linguistics 2001*, Lancaster, Mar. 2001.
- M. Volk. Combining unsupervised and supervised methods for pp attachment disambiguation. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.
- M. Volk. How bad is the problem of PP-attachment? A comparison of English, German and Swedish. In *Proceedings of ACL-SIGSEM Workshop on Prepositions*, Trento, Apr. 2006.
- J. Zavrel, W. Daelemans, and J. Veenstra. Resolving PP attachments ambiguities with memory-based learning. In *Proceedings of Computational Linguistics in the Netherlands*, pages 207–221, Eindhoven, Netherlands, 1996.
- S. Zhao and D. Lin. A Nearest-Neighbor Method for Resolving PP-Attachment Ambiguity. In *Proceedings of the First International Joint Conference on Natural Language Processing*, 2004.