Squib

Prepositional Phrase Attachment without Oracles

Michaela Atterer*
University of Stuttgart

Hinrich Schütze**
University of Stuttgart

Work on prepositional phrase (PP) attachment resolution generally assumes that there is an oracle that provides the two hypothesized structures that we want to choose between. The information that there are two possible attachment sites and the information about the lexical heads of those phrases is usually extracted from gold-standard parse trees. We show that the performance of reattachment methods is higher with such an oracle than without. Because oracles are not available in NLP applications, this indicates that the current evaluation methodology for PP attachment does not produce realistic performance numbers. We argue that PP attachment should not be evaluated in isolation, but instead as an integral component of a parsing system, without using information from the gold-standard oracle.

1. Introduction

One of the main challenges in natural language parsing is the resolution of ambiguity. One frequently studied type of ambiguity is prepositional phrase (PP) attachment. Given the quadruple (v,n1,p,n2), where v is the head of a verb phrase, n1 is the head of an NP1 dominated by v, p is the head of a prepositional phrase, and n2 the head of an NP2 embedded in the PP, the task of PP attachment is to determine whether we should attach the PP to the verb v or the noun n1 (Hindle and Rooth 1993).[1]

Work on PP attachment resolution generally assumes that there is an oracle that provides the quadruple (v,n1,p,n2), where we define an oracle as a mechanism that provides information that is not present in the data in their naturally occurring form. In PP attachment, the oracle is usually implemented by extracting the quadruple (v,n1,p,n2) from the gold-standard parse trees. In an application, a PP attachment module would be used to change the attachment of prepositional phrases in preliminary syntactic analyses produced by a parser—or **reattach** them. The problem with oracle-based work on PP attachment is that when the parser does not find the gold-standard NP1 or PP, for instance, the attachment ambiguity is not recognized, and the correct solution cannot

* Institute for Natural Language Processing, University of Stuttgart, Azenbergstr. 12, 70174 Stuttgart, Germany. E-mail: atterer@ims.uni-stuttgart.de.
** Institute for Natural Language Processing, University of Stuttgart, Azenbergstr. 12, 70174 Stuttgart, Germany. E-mail: hinrich@hotmail.com.

1 PP attachment ambiguities also occur in constructions other than V NP PP (Mitchell 2004). These cases are less frequent and in our opinion do not affect the main conclusions of this article.

Submission received: 13 January 2007, accepted for publication: 7 May 2007.

Table 1
Performance of Collins and Brooks (C&B), Olteanu and Moldovan (O&M) and Toutanova, Manning, and Ng (TM&N) on RRR when no resources other than the training set are used.

Algorithm	Accuracy (%)
C&B	84.18
O&M	84.60
TM&N	85.86

be found by the PP attachment method. Likewise, it is harder for the reattachment method to find the correct attachment if the heads n1 or n2 are not correctly identified by the parser. The oracle approach to PP attachment differs from many other tasks in NLP like part-of-speech tagging or parsing, in which usually no data based on manual annotation are part of the input. Thus, published evaluation results for tagging and parsing accurately reflect the performance we would expect in an application whereas PP reattachment results arguably do not.

We will show in this article that the performance of reattachment methods is higher when an oracle is available. This means that the current evaluation methodology for PP attachment does not produce realistic performance numbers. In particular, it is not clear whether reattachment methods improve the parses of state-of-the-art parsers. This argues for a new evaluation methodology for PP attachment, one that does not assume that an oracle is available.

To create a more realistic setup for PP reattachment, we replace the oracle with Bikel's parser (Bikel 2004). With the removal of the oracle and the introduction of the parser, the baseline performance also changes. It is no longer the performance of always choosing the most frequent attachment. Instead, it is the attachment performance of the parser. In fact, we will see that it is surprisingly difficult for reattachment methods to beat this baseline performance.

The fact that standard parsers perform well on PP attachment also prompted us to investigate the performance of a standard parser on traditional oracle-based PP-attachment. We find that standard parsers do well, further questioning the soundness of the traditional evaluation methodology.

We compare our results with three other approaches: Collins and Brooks (1995; henceforth C&B), Olteanu and Moldovan (2005; henceforth O&M), and Toutanova, Manning, and Ng (2004; henceforth TM&N). We call these three methods PP **reattachers**. We selected these three methods because they perform best on the widely used PP attachment evaluation set created by Ratnaparkhi, Reynar, and Roukos (1994). We call this data set **RRR**. RRR consists of 20,801 training and 3,097 test quadruples of the form (v,n1,p,n2). The performance of the three reattachers on RRR is shown in Table 1.[2]

This article is structured as follows. Section 2 compares the performance of the reattachers on oracle-based reattachment with that of a standard parser (using artificial sentences built from the RRR set) and finds no significant difference in performance. In Section 3, we look at reattachment without oracles (using the Penn Treebank) and argue that realistic performance numbers can only be produced in experiments without oracles. Sections 4 and 5 discuss the experiments and related work. Section 6 concludes.

2 The table shows the performance of the algorithms when only treebank information is used and no information from other resources, such as WordNet.

2. Experiment 1

2.1 Method

We use the Bikel parser (Bikel 2004) with default settings for training and parsing. In Experiment 1, we convert the RRR set into artificial sentences. The data consist of quadruples of the form (v,n1,p,n2): verb, noun, preposition, and embedded noun. To create sentences, we add the generic subject *they*. For example, the quadruple *join board as director V* becomes:

```
( (S (NP (PRP They) ) (VP (VB join) (NP (NN board) ) (PP (IN as)
(NN director) ) ) (. .) ) )
```

Note that these artificial sentences are not necessarily grammatically correct. For example, subject and verb need not agree.

When we train on these artificial sentences, attachment decisions are independent of the embedded noun n2 because the Bikel parser does not percolate non-head information up the tree. We therefore call this experiment **bilexical** because we only use standard lexical head–head relationships. To simulate a parser that takes into account all four elements of the quadruple, we annotate the preposition with the embedded noun:

```
( (S (NP (PRP They) ) (VP (VB has) (NP (NP (NN revenue) )
(PP (IN ofmillion) (NN million) ) ) ) (. .) ) )
```

We call this mode **trilexical** because we effectively model three-word dependencies (e.g., *have-of-million* vs. *revenue-of-million*). To avoid problems with sparseness, we restrict the annotation to the N most frequent n2 nouns of the RRR train data. We chose $N = 20$ based on the performance on the development set.[3]

2.2 Results

Table 2 shows the accuracy of Bikel's parser in bilexical and trilexical mode. The parser sometimes does not recognize the attachment ambiguity. For example, it may fail to correctly identify the PP. We call these cases NAs (non-attachment cases) and compute three different evaluation measures:

NA-error	NAs are counted as errors.
NA-default	Noun attachment (the more frequent attachment) is chosen for NAs.
NA-disc(ard)	NAs are discarded from the evaluation.

We assume that a reattacher like C&B could only be applied to sentences where the ambiguity is recognized, namely, non-NA cases. It would then change the attachment decision in these cases. After reattaching, for this particular set of cases, the accuracy would correspond to the reattachment method's accuracy. We are interested in whether the reattacher is able to beat the performance of the parser we use.

3 On the development set, performance dropped around 0.5 percentage points when using $N = 10$, 0.3 percentage points when using $N = 30$, and 0.9 percentage points when using $N = 50$, for instance.

Table 2
Accuracy (Acc) in percent of the Bikel parser on RRR. For NA-discard, the size of the test set is indicated.

	NA	Acc	C&B	O&M	TM&N
			Significantly different?		
bilex	error	83.18	no	no	yes*
bilex	default	83.44	no	no	yes*
bilex	disc; 3,082 trees	83.58	no	no	yes
trilex	error	83.73	no	no	yes
trilex	default	83.98	no	no	yes
trilex	**disc; 3,082 trees**	**84.13**	**no**	**no**	**no**

Note: The last three columns show whether differences to C&B, O&M, and TM&N are significant at the 0.05 level ("yes") or 0.01 level ("yes*") (χ^2 test).

The Bikel parser's performance (without changing attachments) is slightly lower than C&B's, O&M's, and TM&N's. However, for the trilexical case, the difference is not statistically significant for NA-discard. We consider NA-discard to be the most realistic evaluation measure, because when integrated into a parser, reattachers can only correct the attachment in those sentences where the attachment ambiguity was correctly identified. The parser often identifies one of the possible attachments correctly, but not the other. The reattacher cannot operate on these sentences.

The significance for the NA-discard case was calculated using the results of O&M and TM&N on the same 3,082 sentences (which O&M and TM&N kindly provided to us) and the results of our implementation of the C&B-algorithm.

3. Experiment 2

The setup in Experiment 1 is typical of work on PP attachment, but it is not a realistic model of PP attachment in practice. In Experiment 2, we look at reattachment in naturally occurring sentences if an oracle that identifies the two alternative attachments is not available.

We first tried to convert RRR into a set of sentences by identifying for each quadruple the sentence in the Penn Treebank (PTB) 0.5 it was extracted from. However, we were not able to train the Bikel parser on this set because of inconsistencies and other problems (truncated incomplete sentences, etc.) with the 0.5 treebank. On the other hand, it was not straightforward to create training, test, and development sets from PTB 3, which contains the corresponding sentences. Due to changes in the treebank versions, a considerable number of the quadruples was missing in PTB 3.

For these reasons, we identified as our test data the 3,097 PTB 0.5 sentences the RRR quadruples had been extracted from because for testing we only need string inputs (and the gold-standard attachment annotations of the quadruples).

Our training data consist of those 45,156 PTB 3 sentences that remain of the total number of 49,208 PTB 3 sentences after removing (i) the 3,097 test sentences and (ii) PTB 3 sentences that correspond to RRR development set quadruples. (Note that some test sentences had no corresponding sentence in PTB 3.)

We call this evaluation set (45,156 PTB 3 training sentences, 3,097 PTB 0.5 test sentences) **RRR-sent**. The RRR-sent training set contains a mixture of sentences with

and without PP attachment ambiguities. We believe that this is the optimal experimental setup because parsers are usually not trained on a subset of sentences with a particular construction. RRR-sent is available at `http://ifnlp.org/~schuetze/rrr-sent`.

3.1 Method

In bilexical mode, we trained and tested the parser as before except that RRR-sent was used instead of RRR-based artificial sentences. In trilexical mode, we first trained a parser and ran it on the test data to identify the n2 heads of embedded noun phrases. We then head-annotated the prepositions in RRR-sent (both train and test data) using this parser (restricted to the N nouns selected previously). Other prepositions were left unchanged. Note that prepositions also remain unannotated in sentences where the parser does not recognize the embedding PP. We then trained and tested a second instance of the Bikel parser on the head-annotated data.

3.2 Results

Table 3 shows results for Experiment 2. There are many more things that can go wrong in a long natural sentence than in a 5-word artificial one. Thus, the parser recognizes the PP ambiguity less often in this experiment without an oracle than in Experiment 1, where an oracle was available.

There is only a significant difference for NA-error and NA-default (for the latter not in all cases). For NA-discard there is no significant difference.

4. Discussion

In Experiments 1 and 2 the cases in NA-discard are the only ones where an ambiguity could be identified by the parser and hence the only ones where the reattachment methods could actually be employed. But the reattachers fail to considerably improve the parser on these sentences. It is only the absence of an oracle that makes the parser seem worse in the NA-error and NA-default cases. This result shows that PP attachment methods need to be evaluated with respect to whether they can improve a parser.

In the second experiment the results are worse for the trilexical case, that is, when we use information about the noun in the PP. We believe that one reason for this is that we cannot identify the head noun n2 with certainty due to incorrect preliminary parses.

Table 3
Accuracy (Acc) in percent of the Bikel parser on RRR-sent.

NA		Acc	C&B	O&M	TM&N
bilex	error	80.01	yes*	yes*	yes*
bilex	default	82.75	no	no	yes*
bilex	**disc; 2,945 trees**	**84.14**	**no**	**no**	**no**
trilex	error	72.01	yes*	yes*	yes*
trilex	default	79.85	yes*	yes*	yes*
trilex	disc; 2,652 trees	84.09	no	no	no

Note: "yes$*$" marks significance at the 0.01 level. See text for further explanations.

This again points to the methodological problems in previous work on PP attachment: it relies on an oracle that provides the two alternative attachments, including the four heads involved. Results on the traditional task have a tenuous relation at best to performance on PP attachment if no oracle is available.

The results of Experiments 1 and 2 are not directly comparable. A significant number of quadruples (more than 1,000) in the RRR training set do not occur in the RRR-sent training set due to differences between the 0.5 and 3 treebanks. Conversely, at least as many of the quadruples we extracted from the training set in RRR-sent do not match quadruples in RRR.

5. Related Work

Almost all prior work on PP attachment has adopted what we call the oracle-based approach (Stetina and Nagao 1997; Ratnaparkhi 1998; Pantel and Lin 2000; Olteanu 2004; Bharati, Rohini, Vishnu, Bendre, and Sangal 2005), including several recent papers (Calvo, Gelbukh, and Kilgarriff 2005; Merlo and Ferrer 2006; Volk 2006; Srinivas and Bhattacharyya 2007). None of these papers attempt to improve a parser in a realistic parsing situation.

However, a few studies were published recently that do evaluate PP attachment without oracles (Olteanu 2004; Atterer and Schütze 2006; Foth and Menzel 2006). Our experimental results suggest that the evaluation strategy of integrating PP reattachers into a baseline parser should be adopted more widely.

As we only use a parser and no additional resources such as WordNet, dictionaries, Web data, or named-entity recognizers and stemmers, we restrict ourselves to comparing non-oracle reattachment to work that is evaluated using the training data only (and no other resources) as input. Although we have not replicated experiments with additional resources (e.g., Toutanova, Manning, and Ng 2004; Stetina and Nagao 1997), the question as to the significance of oracle-based results for realistic NLP applications arises independently of the use of resources.

6. Conclusion

Our experiments show that oracle-based evaluation of PP attachment can be misleading. Comparing the bilexical and trilexical cases we note the following: when accurate n2 information (i.e., oracle-based information about which word heads NP2) is added in Experiment 1, disambiguation results improve. When noisy n2-information is added in Experiment 2 (in the absence of an oracle), disambiguation results get worse. Similarly, oracle-based disambiguation performance (84.14%) is much higher than unaided performance (80.01%) in Experiment 2.

It is not clear from our results whether or not PP reattachers perform better than state-of-the-art parsers. Although the differences between the best parsing results and the best reattacher results are not significant in Experiments 1 and 2, the reattachers had consistently higher performance. This makes it likely that PP reattachment (performed either by a reattacher or by a reranker [Charniak and Johnson 2005; Collins and Koo 2005] that processes a feature representation with PP attachment information) would improve the parsing results of state-of-the-art parsers. This is a promising direction for future work on PP reattachment.

Our results suggest that the standard oracle-based method for evaluating PP attachment is problematic. It computes numbers that are overly optimistic when compared to

the performance that is to be expected in realistic applications without oracles. Instead, PP reattachers should be directly compared to state-of-the-art parsers to show that they are able to exploit statistical information that current parsing methods ignore. Most importantly, PP reattachers should be tested on the type of data they are intended to process—output from a parser or another module that detects PP attachment ambiguities. Evaluating reattachers on the output of an oracle allows no inferences as to their real performance.

Many advances in NLP in the last decades have been driven by shared tasks, and PP attachment is one of the tasks that has stimulated much interesting research. There are no realistic scenarios in which an oracle for PP attachment ambiguity would be available, however. Given the differences we have found between evaluations with and without oracles, one should think carefully about the experimental framework for PP attachment work one adopts and whether any oracle-based findings carry over to more realistic scenarios without oracles.

Although we are only concerned with PP attachment in this article, similar considerations likely apply to other disambiguation tasks such as attachment of relative clauses and coordinated phrases. In our opinion the evaluation of disambiguation algorithms on oracle-based data should always be justified by stating specifically what has been learned for the non-oracle case.

Acknowledgments

We would like to thank Marian Olteanu and Kristina Toutanova for their evaluation results and Helmut Schmid and the reviewers for valuable comments.

References

Atterer, Michaela and Hinrich Schütze. 2006. The effect of corpus size in combining supervised and unsupervised training for disambiguation. In *Proceedings of the Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006) Main Conference Poster Sessions*, pages 25–32, Sydney, Australia.

Bharati, Akshar, U. Rohini, P. Vishnu, Sushma Bendre, and Rajeev Sangal. 2005. A hybrid approach to single and multiple PP attachment using WordNet. In *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP) 2005*, pages 211–222, Jeju Island, Korea.

Bikel, Daniel M. 2004. Intricacies of Collins parsing model. *Computational Linguistics*, 30(4):479–512.

Calvo, Hiram, Alexander Gelbukh, and Adam Kilgarriff. 2005. Distributional thesaurus vs. WordNet: A comparison of backoff techniques for unsupervised PP attachment. In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005)*, pages 177–188, Mexico City, Mexico.

Charniak, Eugene and Mark Johnson. 2005. Coarse-to-fine *n*-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 173–180, Ann Arbor, MI.

Collins, Michael and James Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 27–38, Somerset, NJ.

Collins, Michael and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

Foth, Kilian A. and Wolfgang Menzel. 2006. The benefit of stochastic PP attachment to a rule-based parser. In *Proceedings of the Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL) 2006 Main Conference Poster Sessions*, pages 223–230, Sydney, Australia.

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

Merlo, Paola and Eva Esteve Ferrer. 2006. The notion of argument in prepositional phrase attachment. *Computational Linguistics*, 32(3):341–378.

Mitchell, Brian. 2004. *Prepositional Phrase Attachment using Machine Learning Algorithms*. Ph.D. thesis, Natural Language Processing Group, Department of Computer Science, University of Sheffield, UK.

Olteanu, Marian and Dan Moldovan. 2005. PP-attachment disambiguation using large context. In *Proceedings of Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 273–280, Vancouver, Canada.

Olteanu, Marian G. 2004. Prepositional phrase attachment ambiguity resolution through a rich syntactic, lexical and semantic set of features applied in support vector machines learner. Masters thesis, University of Texas, Dallas.

Pantel, Patrick and Dekang Lin. 2000. An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 101–108, Hong Kong, China.

Ratnaparkhi, Adwait. 1998. Unsupervised statistical models for prepositional phrase attachment. In *Proceedings of International Conference on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL98)*, pages 1079–1085, Montreal, Canada.

Ratnaparkhi, Adwait, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Workshop on Human Language Technology*, pages 250–255, Plainsboro, NJ.

Srinivas, Medimi and Pushpak Bhattacharyya. 2007. A flexible unsupervised PP-attachment method using semantic information. In *Proceedings of Internationl Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 1677–1682, Hyderabad, India.

Stetina, Jiri and Makoto Nagao. 1997. Corpus based PP attachment ambiguity resolution with a semantic dictionary. In *Proceedings of the Fifth Workshop on Very Large Corpora*, pages 66–80, Hong Kong, China.

Toutanova, Kristina, Christopher D. Manning, and Andrew Y. Ng. 2004. Learning random walk models for inducing word dependency distributions. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML '04)*, pages 815–822, Banff, Alberta, Canada.

Volk, Martin. 2006. How bad is the problem of PP-attachment? A comparison of English, German and Swedish. In *Proceedings of Association for Computational Linguistics—Special Interest Group on Computational Semantics (ACL-SIGSEM) Workshop on Prepositions*, pages 81–88, Trento, Italy.