Combining Dependency Parsing with PP Attachment

Sandra Kübler, Steliana Ivanova
Indiana University
Bloomington, Indiana, USA
skuebler@indiana.edu
stivanov@indiana.edu

Eva Klett
University of Tübingen
Tübingen, Germany
eklett@sfs.uni-tuebingen.de

1 Introduction

Prepositional phrase (PP) attachment is one of the major sources for errors in traditional statistical parsers. The reason for that lies in the type of information necessary for resolving structural ambiguities. For parsing, it is assumed that distributional information of parts-of-speech and phrases is sufficient for disambiguation. For PP attachment, in contrast, lexical information is needed.

The problem of PP attachment has sparked much interest ever since Hindle and Rooth (1993) formulated the problem in a way that can be easily handled by machine learning approaches: In their approach, PP attachment is reduced to the decision between noun and verb attachment; and the relevant information is reduced to the two possible attachment sites (the noun and the verb) and the preposition of the PP. Brill and Resnik (1994) extended the feature set to the now standard 4-tupel also containing the noun inside the PP. Among many publications on the problem of PP attachment, Volk (2001; 2002) describes the only system for German. He uses a combination of supervised and unsupervised methods. The supervised method is based on the back-off model by Collins and Brooks (1995), the unsupervised part consists of heuristics such as "If there is a support verb construction present, choose verb attachment". Volk trains his back-off model on the Negra treebank (Skut et al., 1998) and extracts frequencies for the heuristics from the "Computerzeitung". The latter also serves as test data set. Consequently, it is difficult to compare Volk's results to other results for German, including the results presented here, since not only he uses a combination of supervised and unsupervised learning, but he also performs domain adaptation.

Most of the researchers working on PP attachment seem to be satisfied with a PP attachment system; we have found hardly any work on integrating the results of such approaches into actual parsers. The only exceptions are Mehl et al. (1998) and Foth and Menzel (2006), both working with German data. Mehl et al. report a slight improvement of PP attachment from 475 correct PPs out of 681 PPs for the original parser to 481 PPs. Foth and Menzel report an improvement of overall accuracy from 90.7% to 92.2%. Both integrate statistical attachment preferences into a parser.

First, we will investigate whether dependency parsing, which generally uses lexical information, shows the same performance on PP attachment as an independent PP attachment classifier does. Then we will investigate an approach that allows the integration of PP attachment information into the output of a parser without having to modify the parser: The results of an independent PP attachment classifier are integrated into the parse of a dependency parser for German in a postprocessing step.

2 Data

The data source underlying the experiments reported in the present study were extracted from the Tübingen treebank of Written German, TüBa-D/Z (Telljohann et al., 2005). TüBa-D/Z is a syntactically annotated corpus consisting of newspaper articles from the German newspaper 'die tageszeitung' (taz). At present, it comprises approximately 27 000

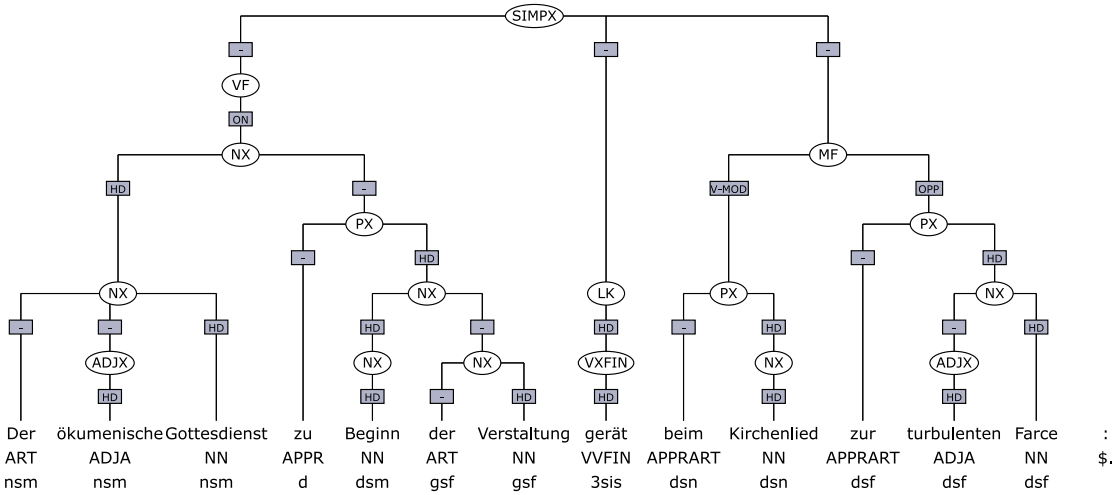

Figure 1: A sample tree from Tüba-D/Z

sentences, or 470 000 words.

For the dependency parsing experiments, the original constituent annotation is converted into dependencies. Since the constituent annotation contains head information on all levels of annotation, the conversion is rule-based rather than heuristic. The few exceptions that do not have head information concern cases of coordination and appositions.

The original tree for sentence (1) is shown in Figure 1, the dependency representation in Figure 2.

(1) Der ökumenische Gottesdienst zu Beginn
The ecumenical service at beginning
der Veranstaltung gerät beim
of the event turns during the
Kirchenlied zur turbulenten Farce.
hymn into a turbulent farce.
”During the hymn, the ecumenical service at the beginning of the event turns into a turbulent farce.’

The PP attachment experiments are based on information from the constituent version of the treebank. For each PP, 11 features are extracted to build the feature vectors: four features contain lexical information for the verb (V), the noun preceding the PP (N1), the preposition of the PP (P) and the core noun of the NP governed by the preposition (N2). Four additional features contain the part-of-speech tags for the respective lexical entries (Vtag, N1tag, Ptag, N2tag). In addition, the syntactic category of N1 (N1cat) is used. The remaining two features contain the distances (i.e. number of words) between the preposition and the verb (|PV|) and the preposition and N1 (|PN|). The last field of each vector contains the correct attachment site for the respective PP. Following the vast majority of approaches to PP-attachment in machine learning, the present study distinguishes only between noun and verb attachments. Note that all PPs that do not clearly attach to a noun are considered to be verb attachments, even though they occasionally modify larger or even missing constituents.

Feature vectors are generated only for PPs in ambiguous positions, i.e. the syntactic annotation constraints of the treebank allow both attachment decisions. In sentence (1), for example, only the last PP, with preposition *zur* is ambiguous, and it results in a feature vector. The first PP precedes the verb, and since it follows an NP in the first position, it unambiguously attaches to the NP. The second PP follows the verb but it does not have a preceding NP, and thus attaches to the verb.

The following guidelines for feature value extraction are applied: The verbal feature values V and

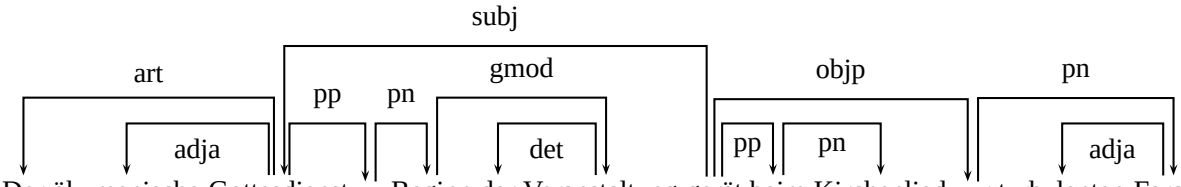

Figure 2: The dependency annotation for sentence (1.)

```
leben,VVFIN,Prozent,NN,ON,unter,APPR,Armutsgrenze,NN,9,6,V
```

Figure 3: The example instance extracted from sentence (2).

Vtag originate from the main verb of the clause holding the ambiguous PP. The nominal feature values N1, N1tag, N2, and N2tag originate from the head words of the NPs in question. For the values of N1, the closest NP to the left of the PP is considered. In the case of verb attachment, the maximal NP is extracted, i.e. as long as the NP node in question is dominated by another NP node, the NP considered for N1 is extended to the higher node. In case of noun attachment, the internal NP shares its mother NP-node with the PP. The syntactic category of the NP considered for N1 is extracted as the value of N1cat. N2 and N2tag are obtained from the head of the phrase governed by the PP. In rare cases, this phrase is an adjectival or adverbial phrase, rather than an NP. The prepositional feature values P and Ptag originate from the preposition of the PP with an ambiguous attachment site. The distance features are obtained by counting the number of words between P and V or P and N1 respectively. The extraction procedure is illustrated for the example sentence in (2).

(2) Heute [VP [V leben] [NP [NP etwa 75
Today live approx. 75
Prozent] [NP der 18,5 Millionen
percent of the 18.5 million
BürgerInnen Nepals]] [PP unter [NP der
citizens of Nepal under the
offiziellen Armutsgrenze]]] .
official poverty threshold .
'Approx. 75 % of the 18.5 mio. citizens of Nepal exist under the official poverty threshold.'

The resulting feature vector is shown in Figure 3. The first two feature values (V and Vtag) relate to the main verb *leben*. For the third and forth value (N1 and N1tag), the head noun and its POS-tag of the complex NP *etwa 75 Prozent der 18,5 Millionen BürgerInnen Nepals* preceding the PP are extraced. The syntactic category (N1cat) of this complex NP is ON (subject). *unter* is the preposition P with its tag APPR (Ptag). Both belong to the ambiguous PP *unter der offiziellen Arbeitsgrenze*. The values for N2 and N2tag originate from the lexical head (*Armutsgrenze*) and the respective POS-tag (NN) in the NP *der offiziellen Arbeitsgrenze*. The distance |PV| is 9 and |PN| is 6. The PP is attached to the verb, as indicated by the class label V in the last field of the feature vector.

3 PP Attachment Experiments

For the PP attachment experiments, the memory-based learner (MBL) TiMBL (Daelemans et al., 2004) was used. MBL is a supervised machine learning approach that stores all training examples in memory and makes new classifications for a new example based on the k nearest neighbors in memory. It has been shown that MBL is well-suited for NLP problems where lexical information is important.

For solving PP-attachment ambiguities, TiMBL stores all available feature vectors in memory. The format of these feature vectors is described in Section 2. During testing, TiMBL classifies unseen fea-

ture vectors. The task is to predict the correct class label for these new examples. The assigned class label indicates either verb or noun attachment for the PP under investigation.

TiMBL provides several different methods for locating the k nearest neighbors in memory. The following choices proved to be optimal for PP attachment in German and for the feature vectors described in Section 2. The algorithm leading to the best results is IB1, which essentially works like the original k-NN algorithm (Aha et al., 1991), except that it searches for the k nearest distances rather than the k nearest neighbors. This means that for each distance, all neighbors with the same nearest particular distance to the new example are selected. The Overlap Metric was chosen as distance metric. It defines the distance between two vectors as the number of mismatching feature values. The weighting of features according to their relevance was accomplished by the weighting method GainRatio. This method bases feature weights on the entropy of the class distribution. Each feature is assigned a weight according to the difference in uncertainty concerning the class labels with and without knowledge of the feature's value. These weights are normalized by the entropy of the feature's values in order to avoid an overestimation of features with many values. From several different methods for handling the class voting procedure, Inverse-Linear Distance proved to give the best results. It assigns a weight of 1 to the nearest neighbor and a weight of 0 to the furthest neighbor. Based on their distances, all other neighbors receive a linearly scaled weight within this interval. With the above combination of parameters and a set of 8 nearest neighbors (k = 8), the best overall accuracy of 81.4% correctly attached PPs was achieved. In order to examine the suitability of the applied feature set, a backward selection was performed. Starting with the full feature vectors, features are successively removed from the feature set until the results can no longer be improved. The present study proves that none of the eleven features are redundant, i.e. all eliminations lead to a loss in attachment accuracy. Thus, the feature set is minimal.

4 Dependency Parsing

For dependency parsing, MALTParser (Nivre, 2006) was used. MaltParser is an implementation of deterministic inductive dependency parsing, based on a memory-based or a Support Vector Machines classifier. An extensive study (Nivre et al., 2007) that tested the parser on 10 different languages showed that its approach is language-independent and reaches state-of-the-art results.

The system consists of 3 main parts: Parser, Guide, and Learner, which operate in two phases, training and parsing. In the training phase, the parser proceeds by parsing sentences from the training corpus, extracting a set of features at every step, and storing them as parse instances. From the collection of parse instances, the learner creates a classifier model, which is invoked by the parser during the parsing phase.

MaltParser supports two parsing algorithms: Nivre's algorithm (used in the experiments for this study) and Covington's incremental algorithms for non-projective parsing. Nivre's algorithm is an adaptation of the standard shift-reduce paradigm, and it looks at two tokens at a time, deciding whether there is a dependency relation between them. The input is represented as a string, and the partially processed structures are stored in a stack. At every step the parser applies one of four possible transitions: *Left-Arc(r)* creates a left dependency relation with label *r*, such that the current token on the stack is a dependent of the next input token. This transition pops the current token from the stack. *Right-Arc(r)* creates a right dependency relation with label *r*, such that the next input token is the dependent of the current token on the stack. This pushes the input token onto the stack. *Reduce* is a transition that pops the top item from the stack when the item has been assigned a head and all dependents. *Shift* takes the next token from the input and pushes it onto the stack.

The Guide allows the user to specify which features are to be extracted from the corpus and to be used in the classifier model. The features are restricted to dependency features, part-of-speech features, and lexical features. For the experiments here, we use the optimal features for German as described in (Nivre et al., 2007): the parser has access to the POS tags of the current word on the stack and the

POS tags of the next input word, as well as to the POS tags of the next three words on the stack and in the input. Additional features are the dependency types of the current token on the stack, of its left- and rightmost dependents, and of the leftmost dependent of the input token. The only lexical features used are the word forms of the top token on the stack and of the input token.

The learner supports a memory-based learning model (TiMBL, see Section 3) and a Support Vector Machines model (LIBSVM) (Chang and Lin, 2001) as underlying classifiers. Our experiments use the TiMBL model. The following parameter settings were used: k was set to 5, Modified Difference Value Metric (MVDM) was used as distance metric down to l=3 (the fall-back distance metric being the Overlap metric for lower frequencies). MVDM is defined as the difference between the conditional probabilities of a class given the particular feature values. No additional feature weighting is used. Class voting was based on Inverse Distance weighting.

5 Combining the Results

A comparison of the results of the PP attachment experiments and the results of the dependency parser for PPs shows that there is room for improvement. For the POS tag APPR for prepositions, unlabeled accuracy of the parser reaches 71.8% (for APPRART, a combination of preposition and determiner, accuracy is 69.5%). The PP attachment module, in contrast, assigns PPs to their head with an accuracy of 81.4%. This difference in results corroborates our first hypothesis that general lexical information is not sufficient for PP attachment. The best results can be obtained having a specialized module that uses only a restricted set of features. In order to combine the results of the two modules, we tested the simplest approach, i.e. using the results of the PP attachment module to correct the output of the dependency parser.

Since the combination of the PP attachment information from the classifier and the dependency parses affects only the dependencies of prepositions, the changes in overall parsing accuracy are necessarily small. In our experiments, the labeled parsing accuracy showed some minor improvement from 86.2% to 86.5%. A closer look at the most frequent label that an outgoing dependency from a preposition is assigned, PP, shows that the improvement is still fairly minor: Accuracy increases from 68.5% to 71.6%. This is only a fraction of the maximal improvement that could be expected, i.e. the accuracy of the PP attachment experiment, 81.4%. The reason why these results are so far below the upper bound lies in the fact that the PP attachment in formation only provides the information on the head of the preposition. However, it does not give any indication of the type of label that the dependency should be assigned. The proof for this can be found in the unlabeled evaluation. Here, the accuracy improves from 71.8% to 77.4% for all words with the POS tag APPR (preposition) and from 69.5% to 75.5% for APPRART (preposition + determiner).

Since the results of a postprocessing integration show only minor improvements, a better solution would be to convert the PP attachment information to a partial dependency annotation and use that as input for the parser. At present, MALTParser cannot handle partially annotated input, but Joakim Nivre (p.c.) announced another version of the parser, which should become available in the near future, and which will be able to handle such input. Once this version is available, we are planning to conduct such preprocessing experiments. Additionally, we are planning to compare these results to results of using the PP attachment information to generate partially bracketed input for constituent parsing with LoPar (Schmid, 2000). We expect that the improvement gained from the PP information will be higher for constituent parsing since a wrong attachment often causes several errors in PARSEVAL evaluation, as shown by Carroll and Briscoe (1996) and Kübler and Telljohann (2002), for example.

References

David Aha, Dennis Kibler, and Marc K. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37–66.

Eric Brill and Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment. In *Proceedings of COLING 94*, Kyoto, Japan.

John Carroll and Ted Briscoe. 1996. Apportioning development effort in a probabilistic LR parsing

system through evaluation. In *Proceedings of the ACL/SIGDAT Conference on Empirical Methods in Natural Language Processing*, pages 92–100, University of Pennsylvania, Philadelphia, PA.

Chih-Chung Chang and Chih-Jen Lin. 2001. LIBSVM: a library for Support Vector Machines. http:www.csie.ntu.edu.tw/∼cjlin/libsvm.

Michael Collins and James Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, Cambridge, MA.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2004. TiMBL: Tilburg memory based learner – version 5.1 – reference guide. Technical Report ILK 04-02, Induction of Linguistic Knowledge, Computational Linguistics, Tilburg University.

Kilian A. Foth and Wolfgang Menzel. 2006. The benefit of stochastic PP attachment to a rule-based parser. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 223–230, Sydney, Australia.

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

Sandra Kübler and Heike Telljohann. 2002. Towards a dependency-based evaluation for partial parsing. In *Proceedings of the LREC-Workshop Beyond PARSEVAL–Towards Improved Evaluation Measures for Parsing Systems*, pages 9–16, Las Palmas, Gran Canaria.

Stephan Mehl, Hagen Langer, and Martin Volk. 1998. Statistische Verfahren zur Zuordnung von Präpositionalphrasen. In B. Schröder, W. Lenders, W. Hess, and Th. Portele, editors, *Computer, Linguistik und Phonetik zwischen Sprache und Sprechen - Tagungsband der 4. Konferenz zur Verarbeitung natürlicher Sprache -Konvens 98*, Universität Bonn.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*. To appear.

Joakim Nivre. 2006. *Inductive Dependency Parsing*. Springer Verlag.

Helmut Schmid. 2000. LoPar: Design and implementation. Technical report, Universität Stuttgart.

Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. 1998. A Linguistically Interpreted Corpus of German Newspaper Texts. In *ESSLLI Workshop on Recent Advances in Corpus Annotation*, Saarbrücken, Germany.

Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, and Heike Zinsmeister, 2005. *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, Germany.

Martin Volk. 2001. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In *Proceedings of Corpus Linguistics 2001*.

Martin Volk. 2002. Combining unsupervised and supervised methods for PP attachment disambiguation. In *Proceedings of COLING-2002*, Taipeh, Taiwan.