Parsers Know Best: German PP Attachment Revisited

Bich-Ngoc Do
Leibniz ScienceCampus
Universität Heidelberg
Heidelberg, Germany
do@cl.uni-heidelberg.de

Ines Rehbein
Data and Web Science Group
Universität Mannheim
Mannheim, Germany
ines@informatik.uni-mannheim.de

Abstract

In the paper, we revisit the PP attachment problem which has been identified as one of the major sources for parser errors and discuss shortcomings of recent work. In particular, we show that using gold information for the extraction of attachment candidates as well as a missing comparison of the system's output to the output of a full syntactic parser leads to an overly optimistic assessment of the results. We address these issues by presenting a realistic evaluation of the potential of different PP attachment systems, using *fully predicted* information as system input. We compare our results against the output of a strong neural parser and show that the full parsing approach is superior to modeling PP attachment disambiguation as a separate task.

1 Introduction

Prepositional phrase (PP) attachment disambiguation, the task of identifying the correct attachment site for each preposition in the syntax tree, has often been described as the canonical case of structural ambiguity in NLP, with crucial impact on semantic interpretation. What makes PP attachment such a challenging task is that *morpho-syntactic* information is often insufficient to resolve the ambiguity, in particular for (semi-)free word order languages, and additional *semantic* information or even *world knowledge* is needed. Even though the PP attachment problem has been studied since the nineties (Hindle and Rooth, 1993; Brill and Resnik, 1994; Ratnaparkhi et al., 1994), it is still one of the hardest problems for syntactic parsing. For constituency parsing of English, Kummerfeld et al. (2012) showed that PP attachment errors are the largest error category across all parsers included in their evaluation. This also holds for dependency parsing. In experiments with the biaffine parser of Dozat and Manning (2017) on German, we found that 21.8% of the unlabeled attachment errors are due to incorrect PP attachments.

To identify the correct attachment site for each preposition, the parser either needs to see these head-PP pairs during training or has to be able to deduce the correct attachment based on seen examples. We can improve lexical coverage by adding more training data, however, treebanking is extremely time-consuming and requires linguistic expertise. In addition, most PPs are adjuncts, which means that they can choose their heads more freely (in comparison to arguments) and it is thus harder to achieve sufficient coverage. A different approach tries to improve PP attachment accuracy by modeling the problem as a separate task. This has the advantage that we do not need fully annotated parse trees as input, and that such a setup makes it easy to integrate a wide range of heterogeneous features. Many studies have tried to solve this task, however, only few have shown that their system is able to improve the output of a strong syntactic parser (see §2).

In the paper, we present a new PP attachment disambiguation system, based on biaffine attention and contextual word embeddings. While obtaining substantial improvements over previous work, we show that modeling all head-dependent pairs jointly (as done in full parsing) allows the system to make more effective use the training data and is thus superior to modeling PP attachment as a separate task.

The paper is structured as follows. We first review statistical methods for PP attachment (§2), focusing on German (§2.1). After outlining some shortcomings of recent work, we reproduce a state-of-the-art

Proceedings of the 28th International Conference on Computational Linguistics, pages 2049–2061
Barcelona, Spain (Online), December 8-13, 2020

PP attachment disambiguation system and evaluate it in a realistic scenario, comparing its performance to that of a strong neural parser (§3). In §4 we propose a new approach based on contextualized word embeddings that overcomes limitations of previous work, and we summarize our results in §5.

2 Related Work

Problem formulation Early work on PP attachment disambiguation (Hindle and Rooth, 1993; Brill and Resnik, 1994; Ratnaparkhi et al., 1994) traditionally formulated the task as a binary choice between a given *verbal* and a *nominal* head candidate while ignoring other parts of speech as possible attachment sites, as well as other potential verbs or nouns in the same sentence that might also be attachment candidates. For example, Brill and Resnik (1994) and Ratnaparkhi et al. (1994) formally define the task input as a quadruple (v, n_1, p, n_2) where v and n_1 are a verbal and a nominal head candidate, p is the preposition and n_2 is the head of the object of p. This setup has been criticized in recent work as highly artificial and unrealistic and has been extended by considering a larger set of possible attachment sites in the sentence, e.g., all words within a certain window (Belinkov et al., 2014), or by selecting head candidates based on linguistic criteria such as topological fields (de Kok et al., 2017a).

Features A diverse set of features has been proposed in the literature. *Lexical association* scores computed on external, large corpora can help to compensate for limited training data in supervised approaches. The unannotated data are parsed to extract tuples of the form (v, n_1, p, n_2), and their counts are used as features for PP attachment disambiguation (Hindle and Rooth, 1993; Ratnaparkhi, 1998; Pantel and Lin, 2000; de Kok et al., 2017b). Alternatively, lexical associations can be estimated based on word co-occurrences in a very large corpus like the World Wide Web (WWW), queried by a search engine (Volk, 2001; Olteanu and Moldovan, 2005).

Semantic information can also help to overcome data sparsity since words that are semantically related are likely to occur in similar contexts, and thus will show similar selectional preferences regarding PP attachment. The semantic word class can be determined based on external resources like WordNet (Fellbaum, 1998) (Brill and Resnik, 1994; Belinkov et al., 2014; Dasigi et al., 2017), or automatically via mutual information clustering (Ratnaparkhi et al., 1994). Alternatively, lexical sparsity can also be tackled by extending the lexicon with semantically similar words from syntactic collocation databases and thesauri (Pantel and Lin, 2000). Semantic information of verbs from FrameNet (Baker et al., 1998) or VerbNet (Schuler, 2005) have also been used as features in PP attachment disambiguation systems (Belinkov et al., 2014; Schuler, 2005).

Syntactic features have been used to recover missing context information in the extracted system input. Examples are the distance between the candidate head and the preposition (Olteanu and Moldovan, 2005; Belinkov et al., 2014; de Kok et al., 2017b), verb subcategorization (Olteanu and Moldovan, 2005) or topological fields (de Kok et al., 2017b). *Pre-trained word embeddings* that capture both syntax and semantics are frequently used as input to PP attachment disambiguation systems with a neural network architecture (Belinkov et al., 2014; Dasigi et al., 2017; de Kok et al., 2017b) to improve lexical coverage and enrich the lexical information.

Comparison with syntactic parsing Early work focuses on evaluating PP attachment disambiguation as an independent task rather than comparing results to parser output or integrating it with parsing. Exceptions are Foth and Menzel (2006) and Roh et al. (2011) who integrate lexical preferences in rule-based parsers. Disambiguation systems that rely on an *oracle* (the mechanism that extracts the two candidate attachment sites, based on gold standard data) have been criticized by Atterer and Schütze (2007). The authors argue that using gold information for candidate extraction is highly unrealistic as it will always include the correct solution in the candidate set, while in real applications the correct solution might not be available. Therefore, PP attachment disambiguation systems should be used to refine the preliminary syntactic analysis of a sentence provided by a parser, or to *reattach* PP attachments. Their experiments with three PP reattachment systems show that none of the systems was able to obtain a significant improvement over the baseline parser. Agirre et al. (2008) later follow their evaluation setup and report a small but significant improvement in parsing accuracy using word sense information.

Up to now, the PP attachment disambiguation problem has been studied extensively, with the latest models incorporating novel techniques like neural networks and word embeddings. Most proposed approaches revolve around improving lexical coverage, either by utilizing large, external corpora or by integrating semantic information. Despite the fact that some systems report better results than a baseline parser, results for PP attachment still fall behind those for state-of-the-art syntactic parsing. In recent work on PP attachment, Dasigi et al. (2017) employ a non-neural network parser with 94.17% accuracy on the Penn Treebank (using gold POS tags) as a baseline parser while the state-of-the-art (without contextualized embeddings) is 96.09% (Zhou and Zhao, 2019). Moreover, while most syntactic parsers perform well with predicted information like POS tags, PP attachment systems are usually evaluated in a setup based on gold information. As a final point, restricting PP attachment sites to nouns and verbs only is very limited compared to full parsing where all possible attachment sites for a PP are considered.

2.1 PP Attachment Disambiguation for German

The focus of our work is on PP attachment disambiguation for German. Notable recent work on German includes de Kok et al. (2017a) and de Kok et al. (2017b) where the former introduces a method for creating a new dataset for PP attachment disambiguation for German, and the latter presents a system and reports results for this dataset.

De Kok et al. (2017a) create their new dataset from the dependency version of the TüBa-D/Z treebank (Hinrichs et al., 2004), aiming at a more realistic setup where multiple potential attachment sites in the sentence are considered, rather than modeling PP attachment disambiguation as a binary classification problem. The authors argue that including *all* nouns and verbs in the sentence as attachment candidates is unnecessary, given that topological word order constraints for German make some positions very unlikely candidates. Instead, they propose to use the topological field model (Drach, 1937; Höhle, 1986), a grammar theory modeling German sentence structure, to extract only those candidates in the sentence that are probable attachment sites, based on the distribution of PPs and their heads across the topological fields annotated in the TüBa-D/Z. Although not emphasized in the paper, de Kok et al. (2017a) consider only nouns and verbs as possible candidates. In addition, the oracle they use to extract candidate heads relies on gold information, i.e., gold POS, topological field tags and syntax trees in the TüBa-D/Z.

De Kok et al. (2017a) present a *neural scoring model* (a feed-forward neural network with one hidden layer) to estimate the probability for a candidate to be the correct attachment site. The input to the system is a triple of *(preposition, prepositional object, head candidate)*. The candidate with the highest score assigned by the scoring model is returned as the correct head of the preposition. The input to the system includes word and POS embeddings, topological field tags for preposition, PP object and candidate; the (absolute and relative) distances between preposition and candidate; and auxiliary distribution scores computed on a large newspaper corpus. Five association scores are computed from both ambiguous and unambiguous triples[1] of preposition, prepositional object and PP head.

Their results are summarized in table 1. The neural scoring model with one-hot vectors for words and POS tags (NN1) achieves only moderate results and is outperformed by nearly 14% when replacing the one-hot vectors with word embeddings (NN2). Including the topological field tags (NN3) and auxiliary distributions (NN4, NN5) as additional features further improves the system's performance.

The success of de Kok et al. (2017b)'s system is based on a combination of well-known techniques for PP attachment disambiguation for English (auxiliary distributions, embeddings), combined with language specific knowledge (topological fields) and modeling (neural networks). However, there are still some shortcomings that need to be addressed. First, no results are reported regarding the relative performance of the system as compared to a strong baseline parser. Second, the fact that system performance is reported on (and, crucially, relies on) gold standard features makes it less attractive in comparison to parsing systems that are able to perform well also with predicted features. This is the motivation behind our work. In the remainder of the paper, we will address these issues by evaluating PP attachment disambiguation systems in a truly realistic setting and assessing their contribution in comparison to a full parsing setup.

[1] Unambiguous cases are those where there is only one possible attachment site for the PP in the sentence.

de Kok et al. (2017b)			Our reproduction	
Name	Model	Accuracy	Name	Accuracy
NN1	NN with one-hot vectors	68.2		
NN2	NN with embeddings	82.0		
NN3	NN2 + topological fields	83.8	PP-REP	84.1
NN4	NN3 + auxiliary all	86.5	PP-REP-AUX	**86.8**
NN5	NN3 + auxiliary unamb.	**86.7**		

Table 1: PP attachment disambiguation results for different settings in the TüBa-D/Z corpus (reproduced from de Kok et al. (2017b)) and our reproduced results.

3 Evaluating PP Attachment in a Realistic Setup

The goal of our experiments is to overcome the limitations outlined above. In particular, we experiment with PP attachment disambiguation in a more realistic scenario where gold information like POS tags and topological field information is not available. Our reference systems are the dependency parser with biaffine attention (Dozat and Manning, 2017) which is among the best systems for parsing German, and the PP attachment disambiguation system from de Kok et al. (2017b) which has been introduced in §2.1.

3.1 Reproducing PP attachment disambiguation results for German

We first try to reproduce the results of de Kok et al. (2017b) described in §2.1, using our re-implementation of their disambiguation system. We report results for the PP attachment dataset (de Kok et al., 2017a) extracted from the TüBa-D/Z, using the same train/test split as in de Kok et al. (2017b). From the 29,033 instances that have been removed from the dataset for training a parser in order to produce auxiliary distributions (de Kok et al., 2017b), we randomly select a development set of size 8,649. However, even when using the same word and POS tag embeddings, we were not able to reproduce the results in de Kok et al. (2017b). Our re-implemented system (without auxiliary scores) achieves an accuracy of only 81.1%, 2.7% below the reported result.

We observe that the cross-entropy loss used to train the system might not be the most suitable choice since it optimizes the score of each candidate independently. We thus replace the cross-entropy loss with the hinge loss which tunes the score of the correct head higher than that of the incorrect one. By changing the loss function and increasing the size of the hidden layer to 1,000, the accuracy of our system increases to 84.1%, which is in the same range as the published results (de Kok et al., 2017b). We call this setting **PP-REP**. For computing the auxiliary distributions, we use articles from the newspaper *taz* from 1986 to 1999 (11.5 mio. sentences, 204.4 mio. tokens), which is a subset of the data used to compute the auxiliary distributions in de Kok et al. (2017b). We parse the corpus with the graph-based parser from the MATE tools[2] (Bohnet, 2010) trained on the German dataset from the CoNLL 2009 Shared Task (Hajič et al., 2009). We keep ambiguous and unambiguous triples and calculate five association scores (similar to the setting NN4 in table 1). With the same hyperparameters as PP-REP, our system with auxiliary distributions achieves an accuracy of 86.8%, slightly higher than the same setup (NN4) (+0.3%) and the best reported result (NN5) (+0.1%) from de Kok et al. (2017b). We refer to this setting as **PP-REP-AUX**. The results of our reproduction study are summarized in table 1.

3.2 Upper bounds for PP attachment disambiguation *without* gold information

After our successful reproduction of previous work, we now proceed to disambiguate PP attachments in a setup that *does not rely on gold POS and topological field tags* and compare its performance with that of the reference parser. In order to do so, the instances of the data used in §3.1 (in the form of preposition, prepositional object and a list of candidate heads and features) have to be mapped back to the original dependency trees to enable a comparison with full parsing. Unfortunately, the process of mapping the triples back to the corresponding treebank trees is not straightforward. First, the format of the PP attachment dataset (de Kok et al., 2017a) does not provide the information needed to trace back

[2]https://code.google.com/p/mate-tools

	POS	TF	P	O	N	V
de Kok et al. (2017a)	gold	gold	gold	gold	gold	gold
Experiment 1 (§3.2)	pred	pred	pred	gold	pred	gold
Experiment 2 (§3.3)	pred	pred	pred	gold/pred	pred	pred

Table 2: Gold/predicted features used in PP attachment experiments. POS: POS tags, TF: topological fields, P: prepositions, O: prepositional objects, N: nominal candidate heads, V: verbal candidate heads.

the training instances to the original trees as it lacks the original structural information (e.g., the position of the words in each instance). Thus, one instance can be mapped to different trees (or different positions in the same tree). Second, the train/test split is done by randomly selecting PP *instances*, not *sentences*. Therefore, when being mapped back, there are trees that appear in both training and test splits. Of course, it would be easy to remove those trees from the training set in order to create non-overlapping sets, but then the results would not be comparable to the parsing results, as those are obtained on a different dataset. For those reasons, we decided to experiment on the German dataset from the SPMRL 2014 Shared Task (Seddah et al., 2014) which is slightly larger than the PP attachment dataset used in de Kok et al. (2017b)[3] but does not contain gold information on topological fields. Thus, in the next step, we will predict topological fields for this dataset.

Reference parser We re-implement the parser of Dozat and Manning (2017) and train it with 100-dimensional dependency-based word embeddings (Levy and Goldberg, 2014).[4] Our model achieves 93.65% unlabeled attachment score (UAS) and 92.22% labeled attachment score (LAS) on the German SPMRL test set using the predicted POS described below.

Predicting POS and Topological Fields Following de Kok and Hinrichs (2016), we model topological field prediction as a sequence labeling task. Rather than predicting deep topological field structures (which are recursive tree structures with potentially nested layers of annotation), we flatten the trees and only annotate each word with its *nearest* topological field tag from the tree hierarchy. Although this method leads to a loss in information when the sentence contains nested fields, it corresponds to the features used in de Kok et al. (2017b). Our topological field labeler is a neural network with two bidirectional LSTM layers and a conditional random field (CRF) decoder. The labeler is trained on the Universal Dependencies (UD) version of the TüBa-D/Z (Çöltekin et al., 2017) which was randomly split into train/dev/test sets with a size of 94,210/5,230/5,347 trees. The POS tags in the SPMRL and TüBa-D/Z datasets represent different interpretations of the same POS tagset (STTS, Schiller et al. (1999)). We thus use MarMoT (Mueller et al., 2013) to train a POS tagger on the training set of the SPMRL dataset[5] and re-tag the TüBa-D/Z. For the SPMRL data, we use MarMoT to assign the POS tags with 10-way jackknifing[6] and predict the topological field tags using our topological field labeler.

Candidate extraction In our first experiment, we study the effect of using *automatically predicted* information for candidate extraction. We replace the gold POS and topological fields with the predicted ones and calculate the *upper bound* accuracy. We follow the rules described in de Kok et al. (2017a) to extract the nominal head candidates for each preposition[7] and use the gold tree to find the verbal candidate (the main verb), rather than using both gold topological fields and gold trees. In summary, prepositions are determined using the predicted POS tags, prepositional objects are determined using the gold trees, nominal candidates are extracted based on the predicted POS and topological field tags, and verbal candidates are extracted based on gold tree information. Table 2 shows the differences between the features used in de Kok et al. (2017b) and in our experiments.

[3]The SPMRL dataset includes 50,000 sentences of German newspaper text, and the extracted PP attachment dataset later contains 45,129 training instances.

[4]The dependency-based embeddings have been trained on the SdeWaC corpus (Faaß and Eckart, 2013).

[5]The POS tagger achieves 97.17% on the SPMRL test set.

[6]The accuracy of the POS tagger on the whole SPMRL dataset is 98.03%.

[7]We follow de Kok et al. (2017a) and consider *adpositions* (both *prepositions* and *postpositions*) in our experiments.

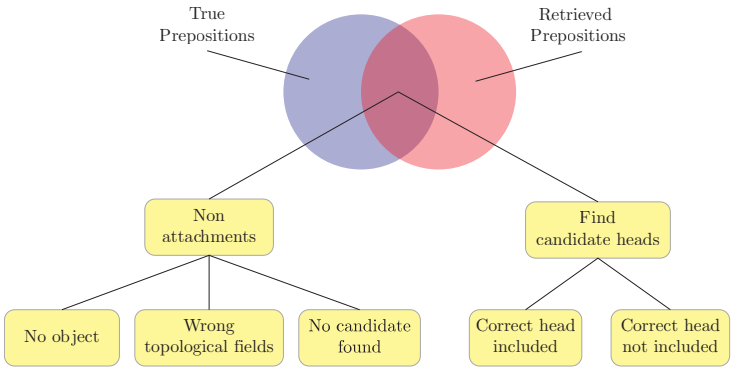

Figure 1: Extracting a PP attachment disambiguation dataset with predicted information.

Upper bound for PP attachment disambiguation We call the set of all gold standard prepositions in the SPMRL test data **PP-SPMRL** (9,273 instances). On the test set of the SPMRL data with predicted POS tags and topological fields, the extraction algorithm finds at least one candidate for 91.66% of the prepositions. The *non-attachments*, where no head candidate was found, are cases where the ambiguity cannot be recognized (because of errors in the predicted information), or are not covered by the extraction rules (e.g., the preposition has no object) (see figure 1). The correct heads are detected in only 82.73% of all cases. This is not only caused by errors in the prediction of topological fields, but also because the algorithm only considers nouns and verbs as possible candidates.[8] This means that the upper bound for recall for PP attachment disambiguation is 82.73% on the test set of the SPMRL data. In comparison, the reference parser achieves 86.17% accuracy for predicting the head of each preposition on the same dataset. If we only consider those 8,360 instances where the gold head of the preposition is either a noun or a verb (we call this set **PP-SPMRL-NV**), the upper bound recall of the disambiguation system increases to 91.56%, whereas the accuracy of the reference parser is 87.21%. Our experiment shows that by limiting head candidates to nouns and verbs only and using predicted POS tags and topological fields, the disambiguation system always performs worse than the parser when evaluating on *all* prepositions. In the next experiment, we will see if the disambiguation system has any advantage over the parser when considering only prepositions with nominal or verbal heads.

3.3 Real-world evaluation of PP attachment disambiguation and PP reattachment

After having established the upper bound performance that we can expect in a realistic scenario, we now assess the performance of the PP attachment disambiguation approach on the output of a strong parser. That is, we are interested in whether the PP attachment disambiguation system can help to improve the performance of the reference parser. Following Atterer and Schütze (2007), we now replace the gold trees used in experiment 1 with the ones predicted by the parser.

The only difference to the previous procedure for extracting candidate heads is that we now rely on the parser's predictions also for finding verbal candidates and prepositional objects, instead of using gold information as before. More specifically, at *test* time, prepositions are identified using the *predicted* POS tags, prepositional objects are determined using the *predicted* parse trees, nominal candidates are extracted based on the *predicted* POS and topological field tags, and verbal candidates are extracted using the *predicted* parse trees. For the train and dev sets, in contrast, only correct instances (prepositions with verbal or nominal heads according to gold information) are considered. All other cases are filtered out as the extraction rules of de Kok et al. (2017a) are restricted to finding nominal and verbal candidates, hence the disambiguation system is only able to select the correct head among nouns and verbs. We further add the correct head to the candidate set in the training and development data if the extraction algorithm fails to include it. The types of features used in this experiment are summarized in table 2.

In comparison to de Kok et al. (2017a), our dataset addresses a more realistic scenario in which the candidate heads for the PP are chosen based on *predicted information*, thus the accuracy for PP attach-

[8]For 9.89% of the prepositions in the SPMRL test set, the head is neither a noun nor a verb.

	PP Attachment						PP Reattachment		
	PP-SPMRL			PP-SPMRL-NV			PP-SPMRL	PP-SPMRL-NV	Parsing
System	P	R	F1	P	R	F1	Accuracy	Accuracy	UAS
Parser	86.29	85.73	86.01	87.33	86.87	87.10	86.17	87.21	93.65
Exp.1 (§3.2)									
Upperbound	100	82.73	90.55	100	91.56	95.59	_	_	_
Exp.2 (§3.3)									
Upperbound	100	80.45	89.17	100	89.03	94.20	_	_	_
PP-REP	77.76	71.22	71.45	84.59	78.80	81.60	_	_	_
all	_	_	_	_	_	_	82.68	85.84	93.29
N&V	_	_	_	_	_	_	84.91	85.81	93.52
PP-REP-AUX	79.52	72.84	76.03	86.52	80.60	83.45	_	_	_
all	_	_	_	_	_	_	83.46	86.63	93.37
N&V	_	_	_	_	_	_	85.59	86.58	93.59

Table 3: PP attachment disambiguation and PP reattachment results on the German SPMRL test set.

ment is bound by the errors from all steps in the pipeline. We train the same system used to reproduce the results of de Kok et al. (2017b) (§3.1) on our newly created dataset for PP attachment disambiguation.

Evaluation metrics for PP attachment disambiguation We follow Agirre et al. (2008) and report *precision*, *recall* and F1 for PP attachment disambiguation. An instance is considered *correct* if its preposition is also a preposition in the gold standard and the system identified the correct head for the preposition. An instance is considered *incorrect* if the preposition is also a preposition in the gold standard but the system assigned an incorrect head. Instances where the preposition is not included in the gold standard are discarded. *Precision* is calculated as the number of correct instances divided by the number of correct and incorrect instances (but ignoring the discarded instances), while *recall* is measured as the number of correct instances divided by the number of prepositions in the gold data. Non-attachment cases are discarded from the evaluation, similar to the `NA-disc(ard)` metric from Atterer and Schütze (2007).

Evaluation metrics for PP reattachment In addition to the PP attachment metrics described above, we also report combined results for the reference parser and the PP attachment disambiguation system (i.e. we *reattach* the PPs predicted by the parser). In the parser output, the head of a preposition is replaced with the one predicted by the disambiguation system in two different ways: `all`: we replace the head of a preposition in the parser output with the one predicted by the disambiguation system for *all* prepositions; `N&V`: we only replace the head of a preposition if the current head predicted by the parser is a noun or a verb. The results for PP reattachment are reported as accuracy for PP attachment, and unlabeled attachment score (UAS) for the whole parse tree. Note that accuracy in PP reattachment corresponds to *all* precision, recall and F1 for PP attachment disambiguation (because the parser predicts a head for all words, the sets for gold and retrieved prepositions are the same).

Results Table 3 shows the performance for PP attachment disambiguation and PP reattachment for different settings.[9] Using the parser's predictions to extract verbal candidate heads (Exp.2) instead of gold information (Exp.1) further reduces the upper bound recall for PP attachment. Neither PP-REP nor PP-REP-AUX could surpass the parser in choosing the correct head for the prepositions. Even when combining these systems with the parser and reattaching the PPs, the accuracies are still behind that of the parser on its own. By restricting the reattachment mechanism to those cases where the head determined by the parser is either a noun or a verb (`N&V`), the accuracy for PP reattachment on the PP-SPMRL set increases by 2%, as this reduces the risk of reattaching the preposition to a nominal or verbal head when the head actually belongs to another part of speech. Adding auxiliary distribution scores to the system (PP-REP-AUX) consistently improves the performance for all settings (over PP-REP).

[9] Our PP-REP and PP-REP-AUX systems were trained on datasets with gold prepositional objects, but we found no difference in performance when training them with predicted objects.

Our experiments show that an independent disambiguation system has no advantage over a full parser in determining the correct head for a preposition when tested in a realistic scenario. First, despite its promising capability in experiments with gold information, the upper bound of the system has already been limited by the error propagation when extracting the candidates based on predicted tags and trees. Second, restricting the coverage of the system by considering only nouns and verbs as possible heads results in a high number of misclassified instances. In contrast, neural network dependency parsers are trained end-to-end on predicted information, thus reducing error propagation. They also make better use of the data by training a common classifier for all head-dependent types, an efficient way to cope with data sparseness.

4 PP Attachment without Restrictions

In the previous section, we have shown that techniques to extract plausible head candidates for German PP attachment disambiguation based on topological fields decreases the upper bound of the system, either by reducing coverage or by error propagation when combining the candidate extraction rules with predicted information at test time. In this section, we will thus consider *all words in a sentence* as possible heads instead of restricting candidate selection to certain nouns and verbs, and focus on techniques that can cope with data sparseness and deal with noisy input data in real-world scenarios.

PP attachment disambiguation with biaffine transformations Our first model is **PP-BIAFFINE**, a PP attachment disambiguation system with biaffine transformations similar to the reference parser of Dozat and Manning (2017). The input to the system consists of words, POS tags and the position of the preposition and its object. Prepositions are identified based on predicted POS tags, and their objects are extracted based on gold standard dependency trees for training and predicted parse trees at test time.[10] Words and tags are converted to embeddings. The score for word h being the head of preposition p with PP object o is computed using a biaffine transformation:

$$s_{\text{biaffine}}(h, p, o) = \mathbf{h}_p^{P\top}\mathbf{W}_1\mathbf{h}_h^H + \mathbf{h}_o^{O\top}\mathbf{W}_2\mathbf{h}_h^H + \mathbf{w}^\top\mathbf{h}_h^H \quad (1)$$

where $\mathbf{h}_i^S$ is the projection S of the encoded representation $\mathbf{h}_i$ (based on bidirectional LSTMs) of word i and $S \in \{H, P, O\}$ corresponds to the head, preposition and PP object projections, respectively. $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices, $\mathbf{w}$ is a weight vector. When using topological field tags (+topo), we also convert them into embeddings and concatenate them with word and POS embeddings to form the token representation. In settings with auxiliary distributions (+aux), the 5 auxiliary scores are combined with the biaffine scores using a linear combination. The pre-trained embeddings for words are the same as used to train the reference parser (§3.2).

We evaluate the potential of PP-BIAFFINE on both tasks, PP attachment disambiguation and PP reattachment (table 4). For PP attachment, PP-BIAFFINE outperforms PP-REP and PP-REP-AUX on the set of all prepositions (PP-SPMRL) by a large margin. On the set with nominal and verbal heads (PP-SPMRL-NV), the precision for PP-BIAFFINE +topo and PP-BIAFFINE +topo, +aux is similar to the one for PP-REP and PP-REP-AUX, but recall is much higher. The main reason for this is that PP-BIAFFINE considers all words as head candidates, while the candidate extraction rules can miss a correct attachment site because of errors in the input. This shows that not restricting the candidate set can lead to *better* performance for PP attachment disambiguation. However, the PP-BIAFFINE system is still not able to outperform the reference parser. Using PP-BIAFFINE to *reattach* the PPs predicted by the parser results in lower scores in comparison to using PP-REP and PP-REP-AUX for reattachment. We hypothesize that the performance of our PP attachment systems is still worse than that of the parser, which means that reattaching more PPs (due to the higher recall of PP-BIAFFINE) only lowers results.

PP attachment disambiguation with contextualized word embeddings Although having similar architectures, the performance of PP-BIAFFINE is still behind that of the full parser. The main reason for this is that the PP attachment disambiguation system has less training data: it is only trained on PP attachments while the parser trains a joint classifier for all attachment types. Instead of using more data, we

[10] We also trained the system on predicted prepositional objects, but using gold objects produced slightly higher results.

	PP Attachment						PP Reattachment		
	PP-SPMRL			PP-SPMRL-NV			PP-SPMRL	PP-SPMRL-NV	Parsing
System	P	R	F1	P	R	F1	Accuracy	Accuracy	UAS
Parser	86.29	85.73	86.01	87.33	86.87	87.10	86.17	87.21	93.65
PP-REP (N&V)	77.76	71.22	71.45	84.59	78.80	81.60	84.91	85.81	93.52
PP-REP-AUX (N&V)	79.52	72.84	76.03	86.52	80.60	83.45	85.59	86.58	93.59
PP-BIAFFINE	83.26	82.72	82.99	84.68	84.23	84.46	83.17	84.58	93.32
+topo	83.48	82.94	83.21	84.79	84.34	84.56	83.38	84.69	93.35
+topo,+aux	85.02	84.47	84.75	86.28	85.83	86.05	84.91	86.17	93.50
PP-BIAFFINE+BERT	87.06	86.50	86.78	88.13	87.67	87.90	86.94	88.01	93.71
+topo	86.87	86.30	86.58	88.07	87.61	87.84	86.75	87.95	93.70
+topo,+aux	87.32	86.75	87.03	88.42	87.95	88.19	87.19	88.30	93.74
Parser+BERT	88.40	87.82	88.11	89.45	88.98	89.22	88.35	89.43	94.43

Table 4: PP attachment disambiguation and PP reattachment results on the German SPMRL test set

propose to improve PP-BIAFFINE by *transfer learning* using BERT (Devlin et al., 2019). BERT is a language model based on multi-layer bidirectional Transformers (Vaswani et al., 2017) that are trained to be sensitive to positional context information, resulting in embeddings that represent *contextualized* word information (*contextualized word embeddings*). Devlin et al. (2019) have shown that the $\text{BERT}_{\text{LARGE}}$ model (with 340M parameters) achieves state-of-the-art results on a wide range of NLP tasks.

We now replace the pre-trained word embeddings in PP-BIAFFINE with embeddings provided by the $\text{BERT}_{\text{BASE}}$ Multilingual Cased model.[11] The rest of the system remains the same. We call this model PP-BIAFFINE+BERT. We do not fine-tune BERT on our data, as our experiments showed that this decreases results over simply using the pretrained word embeddings. The performance of PP-BIAFFINE+BERT is shown in table 4. With the addition of BERT, our model outperforms both PP-BIAFFINE and the reference parser on both tasks, PP attachment disambiguation and PP reattachment. Adding topological field information (PP-BIAFFINE+BERT, +topo) results in slightly worse results, while the addition of both topological fields and auxiliary distributions (PP-BIAFFINE+BERT, +topo, +aux) outperforms all previous models so far. However, the improvement we get is lower than the one we obtained when adding auxiliary scores to PP-BIAFFINE (table 3). This suggests that the information BERT learns from raw text is similar to the one provided by the auxiliary distribution scores.

Parsing with contextualized word embeddings In a similar fashion, we can replace the pre-trained word embeddings in the reference parser with BERT to further improve its performance. Using the same $\text{BERT}_{\text{BASE}}$ Multilingual Cased model without fine-tuning, parsing accuracy increases for both PP attachment (+$\sim$2%) and parsing in general (+0.78%) (Parser+BERT in table 4), and excels the performance of PP-BIAFFINE+BERT, +topo, +aux by $\sim$1%. Again, the shared classifier mechanism has an advantage over the system dedicated to predicting PP attachments only.

Our experiments suggest that parsing systems are in general superior to systems specialized for PP attachment disambiguation. This, however, is only true for high resource languages like English and German where we have enough training data to train a good parser. For low resource languages, on the other hand, acquiring more data for PP attachment disambiguation is much easier than getting more annotated full trees for parser training because PP attachment disambiguation systems only require input in form of triples of (head, preposition, PP object). Moreover, systems for PP attachment disambiguation can utilize data from different treebanks, even if they are based on different underlying linguistic theories as long as they agree on the attachment site of the prepositions.

We reduce the amount of data used to train the reference parser to see when the specialized system has an advantage over the parser. The German SPMRL training set contains 40K sentences and 720K tokens. In the first experiment, we create training datasets that are 25%, 50% and 75% of the original size. The hyperparameters of the parser are kept the same as in previous experiments (§3.2, §3.3)[12]. In the second

[11] The $\text{BERT}_{\text{BASE}}$ Multilingual Cased (110M parameters) was trained on cased text from Wikipedia for 104 languages.

[12] We follow the practice of the first ranked system in the CoNLL 2017 Shared Task on parsing UD treebanks (Dozat et al.,

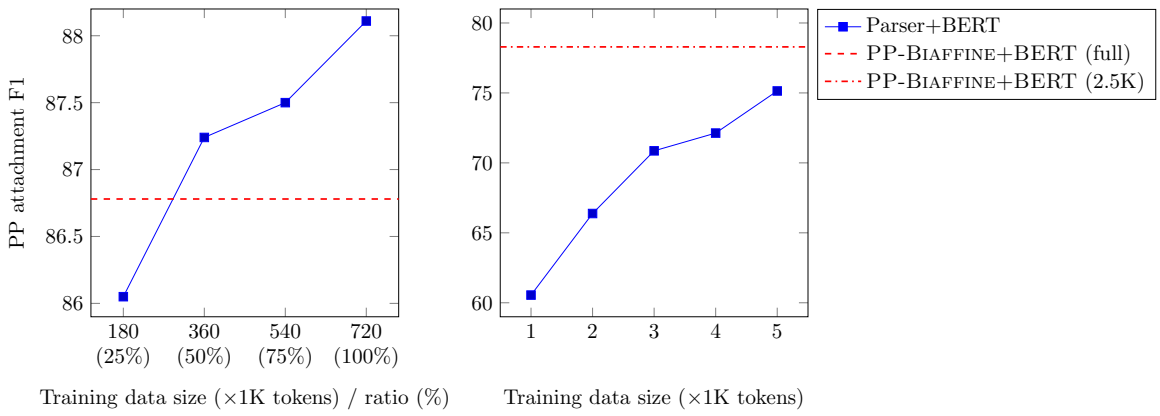

Figure 2: PP attachment disambiguation performance of the reference parser when reducing the size of the training data. Each data point is the average of three different reduced datasets with the same size.

experiment, we simulate a low resource scenario by creating training datasets of sizes ranging from 1,000 to 5,000 tokens. We heuristically reduce the dimensions and number of layers of the reference parser. Likewise, we train the PP-BIAFFINE+BERT[13] system in low resource mode with only 2,500 triples. In both experiments, we use the same POS tags as in previous experiments although POS accuracy could be lower in a real low resourced languages scenario.

The results are illustrated in figure 2. PP-BIAFFINE+BERT trained on the *full* PP attachment data (73K triples, or 146K dependency relations) only outperforms the parser when reducing the training data for the parser to 25% (180K tokens/dependency relations). In the simulated low resource setting, PP-BIAFFINE+BERT trained on 2.5K triples (5K dependency relations) clearly outperforms the parser trained on 5K tokens. The experiments confirm that with the same amount of data annotation, the PP attachment disambiguation system has an advantage over the parser. As the creation of a PP attachment dataset consisting of triples is less expensive than annotating full syntax trees, we believe this could be a way to improve PP attachment accuracy for low resourced languages.

5 Conclusion

We presented an extensive study of the PP attachment disambiguation problem and proposed a new system that combines biaffine attention and pretrained contextualized word embeddings. While our system outperforms recent work on German by a large margin, its performance is still inferior compared to that of a strong neural parser, thus questioning the approach of modeling PP attachment disambiguation as a separate task.

We showed that the lower results for the PP attachment system are caused by error propagation due to using predicted syntactic information for candidate extraction. In addition, the parser can make more efficient use of the training data. While the PP attachment disambiguation system is only trained on the PP attachment edges, the parser makes use of all edge types in the tree to train a joint classifier that predicts the head of each word in a sentence. However, we argue that our system might still be useful for lower resourced languages where, due to a lack of training data, no strong parser is available. While our results are for German, the PP-BIAFFINE+BERT version of our system is language-agnostic and we expect that our findings will carry over to other languages. We leave this for future work.

Acknowledgements

We would like to thank Daniël de Kok for sharing the PP attachment dataset for use in our experiments.

This work was supported in part by the SFB 884 on the Political Economy of Reforms at the University of Mannheim (projects B6 and C4), funded by the German Research Foundation (DFG).

2017).

[13] We assume that there is not enough data to train a good topological field predictor and a good parser for auxiliary distribution scores. For test data, we use the parser trained on 5K tokens to predict the prepositional objects.

References

Eneko Agirre, Timothy Baldwin, and David Martinez. 2008. Improving parsing and PP attachment performance with sense information. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 317–325, Columbus, Ohio, June. Association for Computational Linguistics.

Michaela Atterer and Hinrich Schütze. 2007. Prepositional phrase attachment without oracles. *Computational Linguistics*, 33(4):469–476.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada, August. Association for Computational Linguistics.

Yonatan Belinkov, Tao Lei, Regina Barzilay, and Amir Globerson. 2014. Exploring compositional architectures and word vector representations for prepositional phrase attachment. *Transactions of the Association for Computational Linguistics*, 2:561–572.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China, August. Coling 2010 Organizing Committee.

Eric Brill and Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the 15th International Conference on Computational Linguistics, Volume 2*.

Çağrı Çöltekin, Ben Campbell, Erhard Hinrichs, and Heike Telljohann. 2017. Converting the TüBa-D/Z treebank of German to universal dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 27–37, Gothenburg, Sweden, May. Association for Computational Linguistics.

Pradeep Dasigi, Waleed Ammar, Chris Dyer, and Eduard Hovy. 2017. Ontology-aware token embeddings for prepositional phrase attachment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2089–2098, Vancouver, Canada, July. Association for Computational Linguistics.

Daniël de Kok and Erhard Hinrichs. 2016. Transition-based dependency parsing with topological fields. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–7, Berlin, Germany, August. Association for Computational Linguistics.

Daniël de Kok, Corina Dima, Jianqiang Ma, and Erhard Hinrichs. 2017a. Extracting a PP attachment data set from a German dependency treebank using topological fields. In *Proceedings of the 15th International Workshop on Treebanks and Linguistic Theories (TLT15)*, Bloomington, IN, USA, January.

Daniël de Kok, Jianqiang Ma, Corina Dima, and Erhard Hinrichs. 2017b. PP attachment: Where do we stand? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 311–317, Valencia, Spain, April. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *The 5th International Conference on Learning Representations*, Toulon, France, April.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada, August. Association for Computational Linguistics.

Erich Drach. 1937. *Grundgedanken der deutschen Satzlehre*. Diesterweg.

Gertrud Faaß and Kerstin Eckart. 2013. SdeWaC - A corpus of parsable sentences from the web. In *Language Processing and Knowledge in the Web: Proceedings of the 25th International Conference of the German Society for Computational Linguistics (GSCL 2013)*, pages 61–68, Darmstadt, Germany, September. Springer.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, May.

Kilian A. Foth and Wolfgang Menzel. 2006. The benefit of stochastic PP attachment to a rule-based parser. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics: Main Conference Poster Sessions*, pages 223–230, Sydney, Australia, July. Association for Computational Linguistics.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado, June. Association for Computational Linguistics.

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

Erhard Hinrichs, Sandra Kübler, Karin Naumann, Heike Telljohann, and Julia Trushkina. 2004. Recent developments in linguistic annotations of the TüBa-D/Z treebank. In *Proceedings of the Third Workshop on Treebanks and Linguistic Theories*, pages 51–62, Tübingen, Germany.

Tilman Höhle, 1986. *Der Begriff 'Mittelfeld'. Anmerkungen über die Theorie der topologischen Felder*, pages 329–340. Niemeyer.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the wall street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059, Jeju Island, Korea, July. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.

Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA, October. Association for Computational Linguistics.

Marian Olteanu and Dan Moldovan. 2005. PP-attachment disambiguation using large context. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing*, pages 273–280, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Patrick Pantel 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*, pages 101–108, Hong Kong, October. Association for Computational Linguistics.

Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Workshop on Human Language Technology*, Plainsboro, New Jersey, March.

Adwait Ratnaparkhi. 1998. Statistical models for unsupervised prepositional phrase attachment. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, Volume 2*, pages 1079–1085, Montreal, Quebec, Canada, August. Association for Computational Linguistics.

Yoon-Hyung Roh, Ki-Young Lee, and Young-Gil Kim. 2011. Improving PP attachment disambiguation in a rule-based parser. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, pages 559–566, Singapore, December. Institute of Digital Enhancement of Cognitive Processing, Waseda University.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, IMS, University of Stuttgart and SfS, University of Tübingen.

Karin Kipper Schuler. 2005. *Verbnet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. 2014. Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 103–109, Dublin, Ireland, August. Dublin City University.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Ł. ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5998–6008. Curran Associates, Inc.

Martin Volk. 2001. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In *Proceedings of Corpus Linguistics*, pages 601–606.

Junru Zhou and Hai Zhao. 2019. Head-driven phrase structure grammar parsing on Penn treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408, Florence, Italy, July. Association for Computational Linguistics.