Features for Phrase-Structure Reranking from Dependency Parses

Richárd Farkas, Bernd Bohnet, Helmut Schmid
Institute for Natural Language Processing
University of Stuttgart
{farkas,berndbh,schmid}@ims.uni-stuttgart.de

Abstract

Radically different approaches have been proved to be effective for phrase-structure and dependency parsers in the last decade. Here, we aim to exploit the divergence in these approaches and show the utility of features extracted from the automatic dependency parses of sentences for a discriminative phrase-structure parser. Our experiments show a significant improvement over the state-of-the-art German discriminative constituent parser.

1 Introduction

Both phrase-structure and dependency parsers have developed a lot in the last decade (Nivre et al., 2004; McDonald et al., 2005; Charniak and Johnson, 2005; Huang, 2008). Different approaches have been proved to be effective for these two parsing tasks which has implicated a divergence between techniques used (and a growing gap between researcher communities). In this work, we exploit this divergence and show the added value of features extracted from automatic dependency parses of sentences for a discriminative phrase-structure parser. We report results on German phrase-structure parsing, however, we note that the reverse direction of our approach – i.e. defining features from automatic phrase-structure parses for discriminative dependency parsers – is also manifest which we will address as future work.

Some generative parsing approaches exploited the difference between phrase-structure and dependency parsers. For instance, Klein and Manning (2003) introduced an approach where the objective function is the product of the probabilities of a generative phrase-structure and a dependency parsers. Model 1 of Collins (2003) is based on the dependencies between pairs of head words. On the other hand, the related work on this topic for discriminative parsing is sparse, we are only aware of the following works. Carreras et al. (2008) and Koo et al. (2010) introduced frameworks for joint learning of phrase-structure and dependency parsers and showed improvements on both tasks for English. These frameworks require special formulation of – one or both – parsing approaches while our simple approach allows the usage of arbitrary dependency parsers and any feature-based phrase-structure parser. Wang and Zong (2010) used automatic dependency parses for pruning the chart of a phrase-structure parser and reported a significant improvement. One of our feature templates can be regarded as the generalization of this approach.

2 Feature-Rich Parse Reranking

The most successful supervised phrase-structure parsers are feature-rich discriminative parsers which heavily depend on an underlying PCFG (Charniak and Johnson, 2005; Huang, 2008). These approaches consists of two stages. At the first stage they apply a PCFG to extract possible parses. The full set of possible parses cannot be iterated through in practice, and is usually pruned as a consequence. The *n-best list parser*s keep just the 50-100 best parses according to the PCFG. Other methods remove nodes and hyperedges whose posterior probability is under a predefined threshold from the forest (chart).

The task of the second stage is to select the best parse from the set of possible parses (i.e. rerank this set). These methods employ a large feature set (usually a few millions features) (Collins, 2000; Charniak and Johnson, 2005). The n-best list approaches can straightforwardly employ local and non-local features as well because they decide at the sentence-level (Charniak and Johnson, 2005). Involving non-local features is more complicated in the forest-based approaches. The conditional random field methods usually use only local features (Miyao and Tsujii, 2002; Finkel et al., 2008). Huang (2008) introduced a beam-search and average perceptron-based procedure for incorporating them, however his empirical results show only minor improvement from incorporating non-local features. In this study, we experiment with n-best list reranking and a packed-forest based model as well along with local features exclusively. Our

Proceedings of the 12th International Conference on Parsing Technologies, pages 209–214,
October 5-7, 2011, Dublin City University.

goal is to investigate the extension of the standard feature set of these models by features extracted from the automatic dependency parse of the sentence in question.

3 Dependency Parse-Based Features for Phrase-Structure Parsing

Given the automatic (1-best) dependency parse of the sentence in question, we defined three feature templates for representing hyperedges (i.e. a CFG rule applied over a certain span of words). We illustrate them on two hyperedges E_1 = *(NP die Inseln (PP von Rußland))* and E_2 = *(VP fordern (NP die Inseln) (PP von Rußland))*. Let's assume that the corresponding dependency subtree consists of the following arcs: ROOT→fordern, Inseln$\xrightarrow{DET}$die, fordern$\xrightarrow{OBJA}$Inseln, von$\xrightarrow{PN}$Rußland, fordern$\xrightarrow{PP}$von.

outArc features are counting the dependency arcs which "go out" from the constituent in question. More precisely we count the words within the span whose parent in the dependency tree lays outside the span of words in question. We use the absolute count and the ratio of outArcs among the words of the span. The more arcs go out, the further away is the dependency subtree over the words of the constituent from a dominating subtree. Hence, these features try to capture the "phraseness" of the span of words in question based on the dependency tree. For E_1 we have `outArc=2` and `outArcRatio=2/4` as the parent of *Inseln* and *von* lay outside the constituent. For E_2 we have `outArc=1` and `outArcRatio=1/5`.

POSRel features intend to tune daughter attachments to the dependency parse based on the POS tags of the lexical heads. For this we gather the daughter constituents whose lexical head is linked in the (undirected) dependency tree to the head of the parent constituent. We define features from them using the pair of the two head's POS tag and a triplet using the POS tags and the corresponding dependency label. For E_1 we cannot extract features as the lexical head of the parent (*Inseln*) and the lexical head of the daughter (*von*) are not linked in the dependency tree. For E_2 we have the following binary valued features: `VVFIN-NN`, `VVFIN-NN-OBJA`, `VVFIN-APPR`, `VVFIN-APPR-PP` as both daughter attachments have the corresponding arcs in the dependency tree.

ConstRel features are similar to `POSRel` but use the constituent labels rather than the POS tags of the heads. Thus, once again we do not have any positive feature for E_1, but for E_2 we extract: `VP-NP`, `VP-NP-OBJA`, `VP-PP`, `VP-PP-PP`.

We also investigated the role of case and grammatical functions and extended the `POSRel` and `ConstRel` feature sets by adding this information to the labels. For instance besides `VVFIN-NN-OBJA` and `VP-NP-OBJA` from our example E_2 we also used `VVFIN-NN-Acc-OBJA` and `VP-NP-OA-OBJA`.

Note that the value of `outArc` is 1 iff the word span in question has a dominating dependency subtree in the automatic parse. Wang and Zong (2010) prune hyperedges with `outArc`$\neq 1$ thus this feature can be regarded as a generalization of their approach.

4 Two-Stage Parsing of German

As a **first-stage** parser, we used BitPar (Schmid, 2004), a fast unlexicalized PCFG parser based on a first pass where non-probabilistic bottom-up parsing and top-down pruning is efficiently carried out by storing the chart in bit vectors. Bitpar constructs the probabilistic forest only after top-down pruning, i.e. after computing the posterior probability of each hyperedge given the input sentence. The forest is pruned by deleting hyperedges whose posterior probability is below some threshold.

We used a treebank grammar enriched with case information, lexicalization of selected prepositions, conjunctions, and punctuation symbols, coarse parent category features for adverbs, adverbial phrases, prepositions, PPs and special markers for non-verbal phrases containing a *wh* expression, phrases without a head and clauses without a subject. We applied a second-order markovization of rules below a frequency threshold$_1$, but infrequent second-order Markov symbols are replaced by first-order Markov symbols if the frequency is below threshold$_2$. We used simple regular expressions for unknown word clustering and estimated POS probabilities for unknown words of each cluster based on the word suffix. The relative frequency estimates of the POS probabilities of known words were interpolated with the respective unknown word POS probabilities using

Witten-Bell smoothing. To the best of our knowledge Bitpar with this grammar is the state-of-the-art German generative parser.

At the **second stage**, we used n-best list and forest-based rerankers as well. The feature values of a full possible parse is the sum of the local feature vectors (for the hyperedges) (Charniak and Johnson, 2005). Learning is guided by the so-called oracle parse which is the full parse in the set of possible parses most similar to the gold standard tree. Our oracle extraction method is an extension of Huang (2008)'s dynamic programing procedure which takes into consideration POS tag and grammatical function matches as well and selects hyperedges with higher posterior probability for tie-breaking. For a detailed description of the training and supporting algorithms please refer to Charniak and Johnson (2005) and Huang (2008).

5 Experiments

We evaluate our approach on the Tiger corpora of the Parsing German Shared Task (PaGe) (Kübler, 2008). Its training, development, and test datasets consist of 20894, 2611 and 2611 sentences respectively. We decided to use these corpora to be able to compare our results with other results.

We used the **dependency parser** of Bohnet (2010) to generate the parses for the feature extraction. We selected the parser since it had top scores for German in the CoNLL Shared Task 2009. The parser is a second order dependency parser that models the interaction between siblings as well as grandchildren. The parser was after the Shared Task enhanced by a Hash Kernel, which leads to significantly higher accuracy. We generated the dependency structures by 10-fold cross-validation training of the training corpus. The model for the annotation of the test set and development set was trained on the entire training corpus.

We evaluated the dependency parses themselves in line with PaGe. Table 1 shows the labeled (LAS) and unlabeled attachment scores (UAS) of the dependency parser and compares it with the Malt parser (Nivre et al., 2004; Hall and Nivre, 2008), which was the only and therefore best dependency parser that participated in the PaGe's dependency parsing track. Bohnet's parser reaches higher labeled and unlabeled scores. The last row shows the parsing accuracy with predicted Part-of-Speech. We used the parses with predicted pos tags for our reranking experiments.

Table 1: Dependency parser accuracy. [1]Gold Part-of-Speech tags;[2]Predicted Part-of-Speech tags.

	Test		Dev.	
	UAS	LAS	UAS	LAS
Malt[1]	92.63	90.80	-	-
Bohnet[1]	94.49	92.64	94.80	92.64
Bohnet[2]	93.69	91.71	93.68	91.70

Regarding the **phrase-structure parser**, our grammar extractor used markovization $\text{threshold}_1 = 20$ and $\text{threshold}_2 = 10$ resulting in a grammar with over fifty thousand of rules. Our prior experiments found the forest pruning threshold to be optimal at the order of 10^{-2} which resulted in packed forests with average node number of 108. The oracle scores were 87.1 and 91.4 for the 100-best lists and packed forests, respectively.

At the second stage, we filtered out rare features (which occurred in less than 5 sentences). The new dependency parse-based feature set consists of 9240 and 5359 features before and after filtering. We employed the ranking MaxEnt implementation of the MALLET package (McCallum, 2002) and the average perceptron training of the Joshua package (Li et al., 2009). The update mechanism of the latter one was extended by using the F-score of the candidate full parse against the oracle parse as a loss function (see MIRA (Crammer and Singer, 2003) for the motivation). We used the state-of-the-art feature set of the German phrase-structure parse reranker of Versley and Rehbein (2009) as a baseline feature set. This feature set is rich and consists of features constructed from the lexicalized parse tree and its typed dependencies along with features based on external statistical information (like the clustering of unknown words according to their context of occurrences and PP attachment statistics gathered from the automatic POS tagged DE-WaC corpus, a 1.7G words sample of the German-language WWW). This feature set consists of 1.7 and 0.2 million of features before and after filtering and enables the direct comparison of our results with state-of-the-art discriminative results on German. We use the `evalb` implementation of PARSEVAL as evaluation metric hereafter on basic constituent labels (noGF) and on the conflation of these labels and grammatical functions (GF). We have to mention that our F-values are not comparable to the official results of PaGe – which was our original goal – because the evaluation metric there was a special im-

Table 2: Results achieved by dependency feature-based reranking.

	noGF	GF
Baseline	78.48	66.34
outArc	79.19	67.21
POSRel	79.99	68.13
ConstRel	79.67	67.72
All	80.20	68.32
All+Case	80.35	68.48

plementation for calculating F-value (which differs from evalb for example in handling punctuation marks) and it used gold-standard POS tags in the input (which we thought to be unrealistic). On the other hand, our results are comparable with results of Rafferty and Manning (2008) and Versley and Rehbein (2009).

Table 2 shows the **results** achieved by the MaxEnt 100-best list reranker using one out of the three feature templates alone and their union (`All`) on the development set. `All+Case` refers to the enriched feature set incorporating case information for POS tag and grammatical functions for labels. `Baseline` here refers to the top parse of Bitpar (the first stage parser). We note that the inside probability estimation of Bitpar for an edge is always in our feature set.

Each of the three feature templates achieved significant improvements over a strong baseline – note that our first-stage parser is competitive with Versley and Rehbein (2009)'s two-stage parser – . On the other hand, as the `All` results are just slightly better than `POSRel` (the best individual feature template), the three templates seem to capture similar patterns. The introduction of case information also improved the results, thus we incorporate them into our final feature set. Table 3 illustrates the added value of the dependency features (`Dep=All+Case`) over the reranking feature set of Versley and Rehbein (2009) (`RR`). We also cite here previously published results on the same dataset by Rafferty and Manning (2008) (a generative parser) and Versley and Rehbein (2009) (a conditional random field-based discriminative parser). The rows `RR`, `Dep` and `RR+Dep` show the results achieved by the MaxEnt 100-best list parser while the AvgPer row show the results of the forest-based average perceptron approach using the `RR+Dep` feature set. We report numbers only at this feature configuration due to the lack of space and because the difference between this and n-best list approaches is similarly moderate at other configurations as well.

Table 3: Results achieved by the enriched feature set.

	Develop.		Test	
	noGF	GF	noGF	GF
Rafferty'08	77.40	–	–	–
Versley'09	78.43	67.90	–	–
Baseline	78.48	66.29	79.21	66.63
RR	80.51	68.55	80.95	68.67
Dep	80.35	68.48	80.56	68.39
RR+Dep	81.34	69.73	81.49	69.44
AvgPer	81.41	69.67	81.68	69.42

The results of Table 3 show that our simple features constructed from the automatic dependency parse of the sentence are as useful as the state-of-the-art rich feature set for German. Moreover these two features sets have a certain level of diversity as their union could achieve significantly better results than any of them alone. This is probably due to fact that most of the `RR` features are lexicalized while `Dep` features are unlexicalized. Regarding the two discriminative approaches, our findings are similar to Huang (2008), i.e. the packed forest-based and n-best list procedures achieved similar results by using only local features. We found that the improvements by applying the dependency features are similar at the two evaluation metrics (with and without grammatical functions).

6 Conclusions and Future Work

We presented experimental results on exploiting automatic dependency parses in a discriminative phrase-structure parser. Our simple feature templates achieved around 1.8 points of improvement in terms of F-score over Bitpar, the state-of-the-art generative parser for German and 0.8 when we extended a rich feature set. Although these results are promising, we consider them as the first step on a long road. In the future, we will implement more sophisticated features derived from dependency parses (like dependency paths rather than single edges and non-local ones) and investigate the reverse direction, i.e. whether automatic constituent parses can help dependency parsers.

Acknowledgement

We would like to thank Yannick Versley for his support on reimplementing their feature set and clarifying evaluation issues of German phrase-structure parsing. This work was funded by Deutsche Forschungsgemeinschaft grant SFB 732.

References

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

Xavier Carreras, Michael Collins, and Terry Koo. 2008. Tag, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 9–16.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 173–180.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 175–182.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Comput. Linguist.*, 29:589–637, December.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991.

Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. 2008. Efficient, feature-based, conditional random field parsing. In *Proceedings of ACL-08: HLT*, pages 959–967.

Johan Hall and Joakim Nivre. 2008. A dependency-driven parser for german dependency and constituency representations. In *In Proceedings of the ACL Workshop on Parsing German*, pages 47–54.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL-08: HLT*, pages 586–594.

Dan Klein and Christopher D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Proceedings of Advances in Neural Information Processing Systems*, volume 15.

Terry Koo, Alexander M. Rush, Michael Collins, Tommi Jaakkola, and David Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1288–1298.

Sandra Kübler. 2008. The PaGe 2008 shared task on parsing german. In *Proceedings of the Workshop on Parsing German*, pages 55–63.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren N. G. Thornton, Jonathan Weese, and Omar F. Zaidan. 2009. Joshua: an open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 135–139.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530.

Yusuke Miyao and Jun'ichi Tsujii. 2002. Maximum entropy estimation for feature forests. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 292–297.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In Hwee Tou Ng and Ellen Riloff, editors, *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 49–56.

Anna Rafferty and Christopher D. Manning. 2008. Parsing three German treebanks: Lexicalized and unlexicalized baselines. In *Proceedings of the Workshop on Parsing German*, pages 40–46.

Helmut Schmid. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proceedings of Coling 2004*, pages 162–168.

Yannick Versley and Ines Rehbein. 2009. Scalable discriminative parsing for german. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 134–137.

Zhiguo Wang and Chengqing Zong. 2010. Phrase structure parsing with dependency structure. In *Coling 2010: Posters*, pages 1292–1300.