Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines

Anna N. Rafferty and Christopher D. Manning
Computer Science Department
Stanford University
Stanford, CA 94305
{rafferty,manning}@stanford.edu

Abstract

Previous work on German parsing has provided confusing and conflicting results concerning the difficulty of the task and whether techniques that are useful for English, such as lexicalization, are effective for German. This paper aims to provide some understanding and solid baseline numbers for the task. We examine the performance of three techniques on three treebanks (Negra, Tiger, and TüBa-D/Z): (i) Markovization, (ii) lexicalization, and (iii) state splitting. We additionally explore parsing with the inclusion of grammatical function information. Explicit grammatical functions are important to German language understanding, but they are numerous, and naïvely incorporating them into a parser which assumes a small phrasal category inventory causes large performance reductions due to increasing sparsity.

1 Introduction

Recent papers provide mixed evidence as to whether techniques that increase statistical parsing performance for English also improve German parsing performance (Dubey and Keller, 2003; Kübler et al., 2006). We provide a systematic exploration of this topic to shed light on what techniques might benefit German parsing and show general trends in the relative performance increases for each technique. While these results vary across treebanks, due to differences in annotation schemes as discussed by Kübler (2005), we also find similarities and provide explanations for the trend differences based on the annotation schemes.

We address three parsing techniques: (i) Markovization, (ii) lexicalization, and (iii) state splitting (i.e., subcategorization). These techniques are not independent, and we thus examine how lexicalization and Markovization interact, since lexicalization for German has been the most contentious area in the literature. Many of these techniques have been investigated in other work (Schiehlen, 2004; Dubey, 2004; Dubey, 2005), but, we hope that by consolidating, replicating, improving, and clarifying previous results we can contribute to the re-evaluation of German probabilistic parsing after a somewhat confusing start to initial literature in this area.

One feature of German that differs markedly from English is substantial free word order. This requires the marking of grammatical functions on phrases to indicate their syntactic function in sentences (subject, object, etc.), whereas for English these functions can be derived from configurations (Chomsky, 1965; de Marneffe et al., 2006). While some similar functions are present in English treebanks, they are used more frequently in German treebanks and many more unique functions and category-function pairings exist. Because of the relatively free word ordering in German, the usefulness of parses is substantially increased by generating them with this information. We demonstrate the difficulties introduced by naïvely concatenating these functions to categories and how this treatment interacts with the other parsing techniques. There are several avenues for improving this situation in future work. The versions of the treebanks we use here do not include case information in part-of-speech tags and we do

Treebank	Train	Dev	≤ 40	Test	≤ 40
Tiger	20894	2611	2535	2611	2525
TüBa-D/Z	20894	2611	2611	2611	2611
Negra v2	18602	1000	975	1000	968

Table 1: Size in sentences of treebanks used in this paper. "Tiger" and "TüBa-D/Z" refer to the corpora prepared for the ACL-08 workshop shared task; the full Tiger corpus is much larger. Our Negra results are on the test set.

not use any morphological analyzer; this should be rectified in future work. A new parsing model could be written to treat separate grammatical functions for nodes as first class objects, rather than just concatenating phrasal categories and functions. Finally, assignment of grammatical functions could be left to a separate post-processing phase, which could exploit not only case information inside noun phrases but joint information across the subcategorization frames of predicates.

2 Methodology

We use the Stanford Parser (Klein and Manning, 2003b) for all experiments. An advantage of this parser for baseline experiments is that it provides clean, simple implementations of component models, with many configuration options. We show results in most instances for evaluations both with and without grammatical functions and with and without gold tags. When training and parsing with the inclusion of grammatical functions, we treat each pairing of basic category and grammatical function as one new category. Rules are learned for each such category with a separate orthographic form, with no attempt to learn general rules for nodes with the same basic category but different functions. Clearly, more sophisticated methods of handling grammatical functions exist, but our focus is on providing baseline results that are easily replicable by others.

We focus primarily on the TüBa-D/Z and Tiger corpora, training on the training sets for the ACL 2008 Workshop on Parsing German shared task and providing ablation results based on development set performance. Additionally, we show a limited number of results on the Negra corpus, using the standard training/development/test splits, defined in (Dubey and Keller, 2003). The sizes of these data sets are shown in table 1.

3 Markovization

Previous work has shown that adding vertical Markovization ((grand-)parent annotation) and using horizontal Markovization can greatly improve English parsing performance (Klein and Manning, 2003a). Several papers have already reported partially corresponding results on German: Schiehlen (2004) and Dubey (2004) reported gains of several percent for unlexicalized parsing on Negra; Kübler et al. (2006) agreed with these results for Negra, but suggests that they do not hold for TüBa-D/Z. We extend these results by examining a variety of combinations of Markovization parameters for all three corpora (TüBa-D/Z, Tiger, and Negra) in table 2. No results presented here do include grammatical functions; we present results on the interaction between these functions and Markovization in section 4.

For TüBa-D/Z, we see that adding vertical Markovization provides a substantial performance gain of about 2% (vertical Markovization = 2) for all levels of horizontal Markovization; increasing vertical Markovization improves performance only slightly further. Decreasing horizontal Markovization from the default of infinity for a standard PCFG also provides marginal gains, and decreases the number of rules learned by the parser, creating a more compact grammar. The results of Markovization on the Tiger and Negra corpora illustrate the problems of a large grammar. While a modest improvement is found by using parent annotation (vertical Markovization = 2) when horizontal Markovization is small, increasing either horizontal or vertical Markovization past this point decreases performance due to sparsity. Thus, while the general results concerning Markovization from English hold, the size of performance increase is affected appreciably by the annotation strategy.

In table 3, we show a subset of the results of various Markovization parameters when gold part-of-speech tags are used, focusing on models that performed well without gold tags and that produce relatively compact grammars. Gold tags provide 2–3% absolute improvement in F1 over tagging while parsing; slightly greater improvements are seen when the PCFG model is used individually (3–4% absolute improvement), and absolute improvement does not vary greatly between treebanks. These results are

	TüBa-D/Z			**Tiger**			**Negra**		
Horiz.	Vertical Markov Order			Vertical Markov Order			Vertical Markov Order		
Order	1	2	3	1	2	3	1	2	3
1	86.50	88.60	88.71	76.69	**77.40**	76.46	76.63	**77.20**	75.91
	(+2.76)	*(+1.21)*	*(+0.89)*	*(+3.54)*	*(+3.57)*	*(+3.27)*	*(+2.39)*	*(+2.06)*	*(+2.08)*
2	86.55	88.61	**88.84**	75.91	75.30	74.20	76.39	75.39	73.77
	(+2.63)	*(+1.22)*	*(+0.90)*	*(+3.22)*	*(+3.09)*	*(+3.10)*	*(+3.40)*	*(+2.20)*	*(+2.16)*
3	86.47	88.56	88.74	75.27	74.08	72.88	75.30	74.22	72.53
	(+2.63)	*(+1.18)*	*(+0.90)*	*(+3.36)*	*(+3.41)*	*(+2.85)*	*(+3.74)*	*(+2.12)*	*(+2.60)*
∞	86.04	88.41	88.67	74.44	73.26	71.96	74.48	73.50	71.84
	(+2.17)	*(+1.07)*	*(+0.91)*	*(+3.10)*	*(+3.02)*	*(+2.51)*	*(+3.31)*	*(+1.97)*	*(+3.02)*

Table 2: Factored parsing results for TüBa-D/Z, Tiger, and Negra when tagging is done by the parser. Numbers in italics show difference between factored parser and PCFG, where improvements over the PCFG are positive.

comparable to Maier (2006), which found 3–6% improvement using an unlexicalized PCFG; these absolute improvements hold despite the fact that the Maier (2006) parser has results with 2–4% absolute lower F1 than those in this paper.

4 Inclusion of Grammatical Functions

In this section we examine how the addition of grammatical functions for training and evaluation affects performance. As noted previously, we add grammatical functions simply by concatenating them to the dependent phrasal categories and calling each unique symbol a PCFG nonterminal; this is an obvious way to adapt an existing PCFG parser, but not a sophisticated model of grammatical functions. We also present our shared task results (table 6).

4.1 Effects on Evaluation

As shown in table 4, the inclusion of grammatical functions decreases performance by 10–15% for both treebanks. This is partially due to the increase in grammar size, creating less supporting evidence for each rule, and the fact that the parser must now discriminate amongst more categories. The larger grammar is particularly problematic for Tiger due to its flat annotation style. Adding gold tags (table 5) increases performace by 2–3%, a similar gain to that for the parsers without grammatical functions. We also see that lexicalization provides smaller gains when grammatical functions are included; we discuss this further in section 5. Finally, especially for the Tiger corpus, vertical Markovization diminishes

TüBa-D/Z	Vertical Markov Order	
Horizontal Order	1	2
1	89.66	91.69
	(+1.82)	*(+0.54)*
2	89.72	**91.71**
	(+1.56)	*(+0.43)*
∞	89.34	91.43
	(+1.39)	*(+0.29)*
Tiger	Vertical Markov Order	
Horizontal Order	1	2
1	79.39	**79.67**
	(+2.83)	*(+2.53)*
2	78.60	77.40
	(+2.74)	*(+2.22)*
∞	76.65	75.29
	(+2.50)	*(+1.94)*
Negra	Vertical Markov Order	
Horizontal Order	1	2
1	78.80	**79.51**
	(+2.39)	*(+1.55)*
2	77.92	77.43
	(+2.15)	*(+1.81)*
∞	74.44	73.26
	(+3.10)	*(+3.02)*

Table 3: Factored parsing results for TüBa-D/Z, Tiger, and Negra when gold tags are provided as input to the parser. Numbers in italics show difference between factored parser and PCFG, where improvements over the PCFG are positive.

	TueBa-D/Z		Tiger	
Horiz.	Vertical		Vertical	
Order	1	2	1	2
1	75.97	**77.21**	**60.48**	58.00
	(+2.69)	*(+1.49)*	*(+2.69)*	*(+2.24)*
2		76.96		53.68
		(+1.44)		*(+2.22)*
∞	75.24	76.66	55.36	50.94
	(+2.18)	*(+1.22)*	*(+2.50)*	*(+1.94)*

Table 4: Results for TüBa-D/Z and Tiger when grammatical functions are included and tagging is done by the parser. Numbers in italics show difference between factored parser and PCFG, where improvements over the PCFG are positive.

	TüBa-D/Z		Tiger	
Horiz.	Vertical		Vertical	
Order	1	2	1	2
1	78.91	**80.64**	**67.72**	64.93
	(+1.60)	*(+0.81)*	*(+1.16)*	*(+0.77)*
2		80.32		59.60
		(+0.69)		*(+0.67)*
∞	78.38	80.01	60.36	56.77
	(+1.33)	*(+0.59)*	*(+0.89)*	*(+0.18)*

Table 5: Results for TüBa-D/Z and Tiger when grammatical functions are included and gold tags (including grammatical functions) are given to the parser.

	TüBa-D/Z	Tiger
Petrov & Klein	**83.97**	**69.81**
Rafferty & Manning	79.24	59.44
Hall	75.37	65.18
Rafferty & Manning -gf	73.36	49.03

Table 6: Shared task results (F1) for TüBa-D/Z and Tiger when grammatical functions are included and gold tags are given to the parser. Gold tags include grammatical functions except in the case of "Rafferty & Manning -gf".

performance. Sparsity becomes too great of an issue for increased vertical annotations to be effective: the grammar grows from 11,170 rules with horizontal Markovization = 1, vertical Markovization = 1 to 39,435 rules with horizontal Markovization = ∞, vertical Markovization = 2.

TüBa-D/Z	Fact.		PCFG	
Configuration	F1	Δ	F1	Δ
$H = 1, V = 1$	87.63	+1.63	85.32	+1.58
$H = 1, V = 2$	**88.47**	−0.13	**87.31**	−0.08
$H = 2, V = 2$	88.30	−0.31	87.13	−0.26
$H = \infty, V = 1$	87.23	+1.17	85.27	+1.40
$H = \infty, V = 2$	88.18	−0.23	87.09	−0.25
Tiger	Fact.		PCFG	
Configuration	F1	Δ	F1	Δ
$H = 1, V = 1$	**72.09**	−4.60	**69.09**	−4.06
$H = 1, V = 2$	69.25	−8.15	67.24	−6.59
$H = 2, V = 2$	66.08	−9.22	64.42	−7.79
$H = \infty, V = 1$	67.58	−9.07	64.85	−6.49
$H = \infty, V = 2$	63.54	−11.75	62.21	−8.03

Table 7: Effect of adding grammatical functions information to the training data only. The difference (Δ) is from a parser with same Markovization parameters but not trained with grammatical functions.

4.2 Effects on Training Only

While training and testing with grammatical functions significantly reduces our performance, this does not necessarily mean that we cannot benefit from grammatical functions. We explored whether training with grammatical functions could improve the parser's test time performance on syntactic categories (ignoring grammatical functions), hypothesizing that the functions could provide additional information for disambiguating which rule should be applied. This test also provides evidence of whether decreased performance with grammatical functions is due to sparseness caused by the large grammar or simply that more categorization needs to be done when grammatical functions are included.

We found, as shown in table 7, that grammatical functions provide limited gains for basic categories but have no extra utility once vertical Markovization is added. These results suggest that adding grammatical functions is not only problematic due to increased categorization but because of sparseness (this task has the same categorization demands as parsing without grammatical functions considered in section 3). The Stanford Parser was initially designed under the assumption of a small phrasal category set, and makes no attempts to smooth grammar rule probabilities (smoothing only probabilities

of words having a certain tag and probabilities of dependencies). While this approach is in general not optimal when many category splits are used inside the parser – smoothing helps, cf. Petrov et al. (2006) – it becomes untenable as the category set grows large, multi-faceted, and sparse. This is particularly evident given the results in table 7 that show the precipitous decline in F1 on the Tiger corpus, where the general problems are exacerbated by the flatter annotation style of Tiger.

5 Lexicalization

In the tables in section 3, we showed the utility of lexicalization for German parsing when grammatical functions are not required. This contrasts strongly with the results of (Dubey and Keller, 2003; Dubey, 2004) where no performance increases (indeed, performance decreases) are reported from lexicalization. Lexicalization shows fairly consistent 2–3% gains on the Negra and Tiger treebanks. As the number of tags increases, however, such as when grammatical functions are included, gains from lexicalization are limited due to sparseness. While useful category splits lessen the need for lexicalization, we think the diminishing gain is primarily due to problems resulting from the unsmoothed PCFG model. As the grammar becomes sparser, there are limited opportunities for the lexical dependencies to correct the output of the PCFG grammar under the factored parsing model of Klein and Manning (2003b). Indeed, as shown in table 8, the grammar becomes sufficiently sparse that for many sentences there is no tree on which the PCFG and dependency grammar can agree, and the parser falls back to simply returning the best PCFG parse. This falloff, in addition to overall issues of sparsity, helps explain the drop in performance with the addition of grammatical functions: our possible gain from lexicalized parsing is decreased by the increasing rate of failure for the factored parser. Thus, for future German work to gain from lexicalization, it may be necessary to explore smoothing the grammar or working with a diminished tagset without grammatical functions.

Lexicalized parsing focuses on identifying dependencies. As recognized by Collins (2003), identifying dependencies between words allows for better evaluation of attachment accuracy, diminishing spurious effects on labeled bracketing F1 of different annotation schemes. In particular, Rehbein and van Genabith (2007) correctly emphasize how F1 scores are very dependent on the amount of branching structure in a treebank, and are hence not validly comparable across annotation styles. We evaluate performance on identifying unlabeled dependencies between heads and modifiers, extracting dependencies automatically from the parse trees. Most heads in the TüBa-D/Z and Tiger treebanks are marked, and we use marked heads when possible for training and evaluation. When heads were not marked, we used heuristic rules to identify the likely head. Broadly consistent with the results of Rehbein and van Genabith (2007), Table 9 shows that the disparity in performance between TüBa-D/Z and Tiger is much smaller when measuring dependency accuracy rather than labeled bracketing F1, especially when using gold tags. These results also reverse the trend in our other results that adding grammatical functions greatly reduces F1. While F1 decreases or remains constant when grammatical functions are used with automatic tags, probably reflecting a decrease in accuracy on tags when using grammatical functions, they increase F1 given gold tags. These results suggest both that useful information may be gained from grammatical functions and that the dif-

Dataset	Total Sent.	Parseable w.o. GFs	Parseable with GFs
TüBa-D/Z	2611	2610	2197
Tiger	2535	2534	1592

Table 8: Number of sentences parseable by the factored lexicalized parser. If the factored model fails to return a parse, the parser returns the best PCFG parse, so the parser maintains 100% coverage.

	TüBa-D/Z	**Tiger**
Gold Tags	**91.00**	**90.21**
Auto. Tags	86.90	83.39
Gold Tags -gf	89.89	88.97
Auto. Tags -gf	86.89	85.86

Table 9: Performance (F1) on identifying dependencies in TüBa-D/Z and Tiger. Tags were either provided ("Gold Tags") or generated during parsing ("Auto. Tags"); grammatical functions were used for the first two results and omitted for the final two ("-gf").

ferences between the annotation schemes of TüBa-D/Z and Tiger may not cause as large a fundamental difference in parser performance as suggested in Kübler et al. (2006).

6 Feature Splits

Another technique shown to improve accuracy in English parsing is state splits (Klein and Manning, 2003a). We experimented with such splits in an attempt to show similar utility for German. However, despite trying a number of splits that leveraged observations of useful splits for English as well as information from grammatical functions, we were unable to find any splits that caused significant improvement for German parsing performance. Somewhat more positive results are reported by Schiehlen (2004) – in particular, his relative clause marking adds significantly to performance – although many of the other features he explores also yield little.

7 Errors by Category

In this section, we examine which categories have the most parsing errors and possible reasons for these biases. Two types of error patterns are considered: errors on particularly salient grammatical functions and overall category errors.

7.1 Grammatical Function Errors

A subset of grammatical functions was recognized by Kübler et al. (2006) as particularly important for using parsing results, so we investigated training and testing with the inclusion of these grammatical functions but without any others. These functions were the subject, dative object, and accusative object functions. We found that the three categories had distinctively different patterns of errors, although we unfortunately still do not achieve particularly high F1 for any of the individual pairings of node label and grammatical function. Note that this analysis differs from that of Kübler et al. (2006) due to our analysis of the accuracy of node labels and grammatical functions, rather than only performance on identifying these three grammatical functions (without regards to the correctness of the original node label). Overall, dative objects occur much less frequently than either of the other two types, and accusative objects occur less frequently than subjects. Consistent with sparsity causing degradations in performance, for both Tiger and TüBa-D/Z, we show the best performance on subjects, followed by accusative objects and then dative objects. For all categories, we find that these functions occur most frequently with noun phrases, and we achieve higher performance when pairing tthem with a noun phrase than with any other basic category. While Kübler et al. (2006) suggests these functions are particularly important for parsing, our low performance on dative objects (F1 between 0.00 and 0.06) may not matter a great deal given that dative objects consist of only 0.42% of development set nodes in TüBa-D/Z and 0.76% of such nodes in Tiger.

7.2 Overall Errors

One limiting factor for overall parsing accuracy is roughly defined by the number of local (one-level) trees in the test set that are present in the training set. While changes such as Markovization may allow rules to be learned that do not correspond directly to such local trees, it is unlikely that many such rules will be created. Thus, if a local tree in the test set is not represented in the training set, it is unlikely we will be able to correctly parse this sentence. The number of such local trees and the amount of test set coverage they provide varies widely between TüBa-D/Z and Tiger. Without grammatical functions, the training set for TüBa-D/Z contains 4,532 unique local trees, whereas the training set for Tiger contains 20,957; both have 20,894 complete trees. Local trees from the training set represent 79.6% of the unique local trees in the development set for TüBa-D/Z, whereas they represent 61.8% of unique local trees in Tiger's development set. This translates to 99.3% of total local trees in the development set represented in the training set for TüBa-D/Z versus 92.3% for Tiger. With grammatical functions, the number of unique local trees increases for both TüBa-D/Z and Tiger (10,464 and 32,614 trees in training, respectively), and total coverage in the development sets drop to 98.6% (TüBa-D/Z) and 87.7% (Tiger). Part of the reason for this decrease in coverage with the addition of grammatical functions, and the disparity between corpora, is a large increase in the number of possible categories for each node: from 26 to 139 categories for TüBa-D/Z and from 24 to 192 categories for Tiger.

References

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. In *Computational Linguistics*, pages 589–638.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 449–454.

Amit Dubey and Frank Keller. 2003. Probabilistic parsing for German using sister-head dependencies. In *ACL 41*, pages 96–103.

Amit Dubey. 2004. *Statistical Parsing for German: Modeling Syntactic Properties and Annotation Differences*. Ph.D. thesis, Universitaet des Saarlandes.

Amit Dubey. 2005. What to do when lexicalization fails: parsing German with suffix analysis and smoothing. In *ACL 43*, pages 314–21.

Dan Klein and Christopher D. Manning. 2003a. Accurate unlexicalized parsing. In *ACL 41*, pages 423–430.

Dan Klein and Christopher D. Manning. 2003b. Fast exact inference with a factored model for natural language parsing. *Advances in Neural Information Processing Systems*, 15:3–10.

Sandra Kübler, Erward W. Hinrichs, and Wolfgang Maier˙ 2006. Is it really that difficult to parse German? In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*.

Sandra Kübler. 2005. How do treebank annotation schemes influence parsing results? Or how not to compare apples and oranges. In *Proceedings of RANLP 2005*.

Wolfgang Maier. 2006. Annotation schemes and their inuence on parsing results. In *Proceedings of the COLING/ACL 2006 Student Research Workshop*, pages 19–24.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *ACL 44*, pages 433–440.

Ines Rehbein and Josef van Genabith. 2007. Treebank annotation schemes and parser evaluation for German. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 630–639.

Michael Schiehlen. 2004. Annotation strategies for probabilistic parsing in German. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 390–96.