How Do Treebank Annotation Schemes Influence Parsing Results? Or How Not to Compare Apples And Oranges

Sandra Kübler
Universität Tübingen
Seminar für Sprachwissenschaft
Wilhelmstr. 19
D-72074 Tübingen, Germany
kuebler@sfs.uni-tuebingen.de

Abstract

In the last decade, the Penn treebank has become the standard data set for evaluating parsers. The fact that most parsers are solely evaluated on this specific data set leaves the question unanswered how much these results depend on the annotation scheme of the treebank. In this paper, we will investigate the influence which different decisions in the annotation schemes of treebanks have on parsing. The investigation uses the comparison of similar treebanks of German, NEGRA and TüBa-D/Z, which are subsequently modified to allow a comparison of the differences. The results show that deleted unary nodes and a flat phrase structure have a negative influence on parsing quality while a flat clause structure has a positive influence.

1 Introduction

In the last decade, the Penn treebank (Marcus *et al.* 94) has become the standard data set for evaluating parsers. The fact that most parsers are solely evaluated on this specific data set leaves the question unanswered how much these results depend on the annotation scheme of the treebank. This point becomes more urgent in the light of more recent publications on parsing the Penn treebank such as (Charniak 00; Charniak 01; Klein & Manning 03; Dubey & Keller 03), which show that parsing results can be improved if certain peculiarities of the Penn and the NEGRA treebank annotations are taken into consideration in the probability model. (Klein & Manning 03), e.g., gain approximately 1 point in F-score when they extend POS tag information by the mother node or the lemma. This directly reflects shortcomings in the annotation scheme, which groups prepositions, subordinating conjunctions, and complementizers under the same POS tag. (Charniak 01) reports a 10% reduction in grammar perplexity for his trihead model, which models deeper structure in flat NPs such as “Monday night football”. These findings raise the question whether such shortcomings in the annotation can be avoided during the design of the annotation scheme of a treebank. The question, however, can only be answered if it is known which design decisions are more or less favorable for PCFG parsing.

In this paper, we will investigate how different decisions in the annotation scheme influence parsing results. In order to answer this question, however, a method needs to be developed which allows the comparison of different annotation decisions without comparing unequal categories.

For a comparison of different annotation schemes, one ideally needs one treebank with two different sets of (manual) annotations. An automatic conversion from one annotation scheme to the other is only possible from deeper structures to flatter ones. The other direction would have to be based on heuristics. In this case, there is a high probability that systematic errors are introduced so that only a corrupted annotation in the target annotation scheme will be reached. In the absence of more detailed methods of comparison, testing the effect of modifying individual annotation decisions gives insight into the factors that influence parsing results.

Section 2 gives an overview of treebank pairs for a single language. In section 3, we will describe the treebanks used in this investigation in more detail, section 4 describes the preparatory steps necessary for converting these treebanks into a format that can be treated by a PCFG parser. Section 5 describes the method of comparison, and section 6 discusses the results of the comparison.

2 Comparable Treebanks

For the comparison described above, we need different treebanks which are based on the same language and the same text genre and which are annotated with different annotation schemes. But the annotation schemes must be similar enough to enable a comparison. A comparison between a constituent-based and a dependency-based annotation scheme would be very difficult since, in their original form, they require two different parsing algorithms. A completely determined rule-based conversion between the two is only possible from constituents to dependencies. This is not an optimal solution since decisions in dependency annotations are made on a lexical level and can only

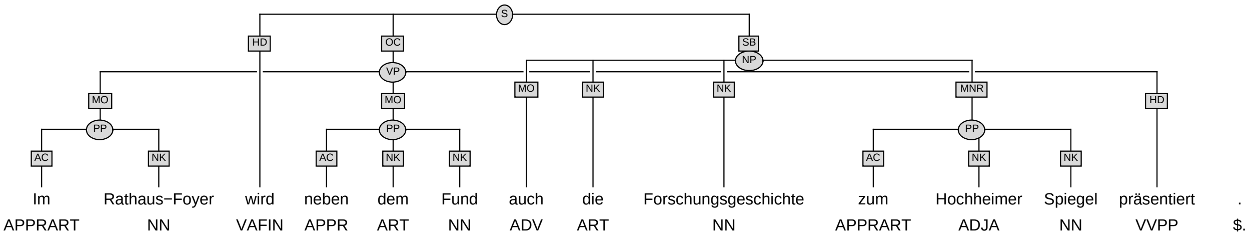

Figure 1: A sample tree from the NEGRA treebank.

be generalized to a certain extent.

One of the very few examples of two treebanks for one language are the Penn treebank (Marcus *et al.* 94) and the SUSANNE corpus (Sampson 93) for English. However, there are significant differences in the size of the treebanks and in the text genres, which make a comparison of the two annotation schemes unfeasible. Another example of such a pair are the two treebanks for Italian, ISST (Montegmagni *et al.* 00) and TUT (Bosco *et al.* 00). ISST uses a constituent-based annotating scheme augmented with grammatical functions; TUT, in contrast, is annotated with dependency relations. For the reason given above, this would not allow a comparison based on constituents. Additionally, both treebanks are of a very restricted size, which makes data sparseness problems very likely.

Only recently, a new pair of treebanks for German has become available, the NEGRA (Skut *et al.* 97) and the TüBa-D/Z (Telljohann *et al.* 04) treebanks. Both treebanks are based on newspaper text, both use the STTS POS tagset (Thielen & Schiller 94), and both use an annotation scheme based on constituent structure augmented with grammatical functions. However, they differ in other respects, which makes them ideally suited for an investigation on how decisions in the design of an annotation scheme influence parsing accuracy.

3 The NEGRA and the TüBa-D/Z Treebanks

Both treebanks use German newspapers as their data source: the Frankfurter Rundschau newspaper for NEGRA and the ’die tageszeitung’ (taz) newspaper for TüBa-D/Z. NEGRA comprises 20 000 sentences, TüBa-D/Z 15 000 sentences. Both treebanks use an annotation framework that is based on phrase structure grammar and that is enhanced by a level of predicate-argument structure. Annotation for both was performed semi-automatically. Despite all these similarities, the treebank annotations differ in four important aspects: 1) NEGRA does not allow unary branching while TüBa-D/Z does; 2) in NEGRA, phrases receive a flat annotation while TüBa-D/Z uses phrase internal structure; 3) NEGRA uses crossing branches to represent long-distance relationships while TüBa-D/Z uses a pure tree structure combined with functional labels to encode this information; 4) NEGRA encodes grammatical functions in a combination of structural and functional labeling while TüBa-D/Z uses a combination of topological fields (Drach 37; Höhle 86) and functional labels, which results in a flatter structure on the clausal level. The two treebanks also use different notions of grammatical functions: TüBa-D/Z defines 36 grammatical functions covering head and non-head information, as well as subcategorization for complements and modifiers. NEGRA utilizes 48 grammatical functions. Apart from commonly accepted grammatical functions, such as *SB* (subject) or *OA* (accusative object), NEGRA grammatical functions also comprise a more extended notion, e.g. *RE* (repeated element) or *RC* (relative clause) [1]. The difference in grammatical functions, however, is difficult to compare since this can only be done in a task-based evaluation within an application that uses these grammatical functions as input.

Figure 1 shows a typical tree from the NEGRA treebank. The syntactic categories are shown in circular nodes, the grammatical functions as edge labels in square boxes. The prepositional phrase “Im Rathaus-Foyer” (in the foyer of the town hall) and the noun phrase “auch die Forschungsgeschichte zum Hochheimer Spiegel” (also the research history of the Hochheimer Spiegel) do not contain internal structure, the noun kernel elements are marked via the functional labels *NK*. The fronted PP is grouped under the verb phrase, resulting in crossing branches. Figure 2 shows a typical example from TüBa-D/Z. Here, the complex noun phrase “Der Autokonvoi mit den Probenbesuchern” (the car convoy with the visitors of the rehearsal) contains a noun phrase and the prepositional phrase with an internal noun phrase, with

[1] For a more detailed comparison of Tüba-D/Z and TIGER, the successor of NEGRA, cf. (Telljohann *et al.* 04).

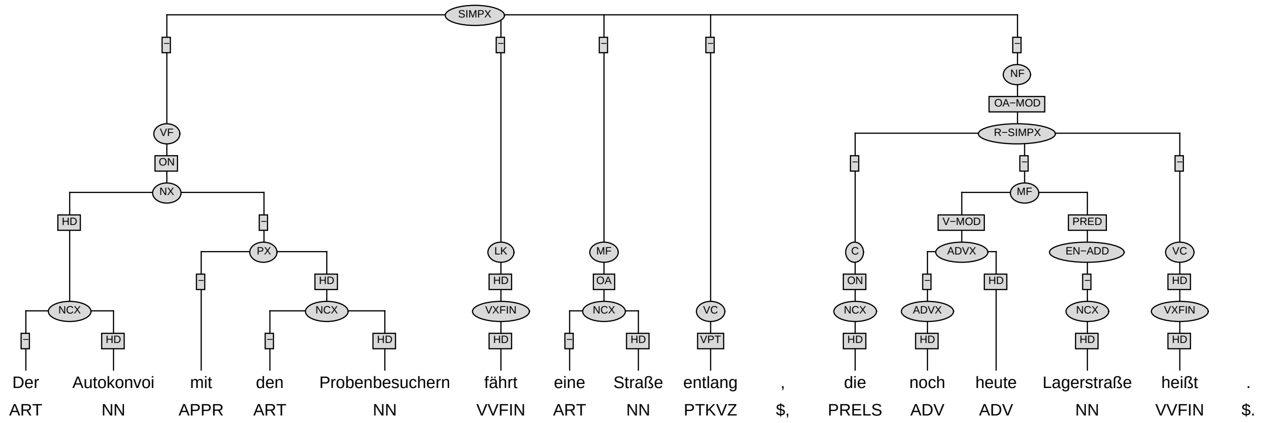

Figure 2: A sample tree from the TüBa-D/Z treebank.

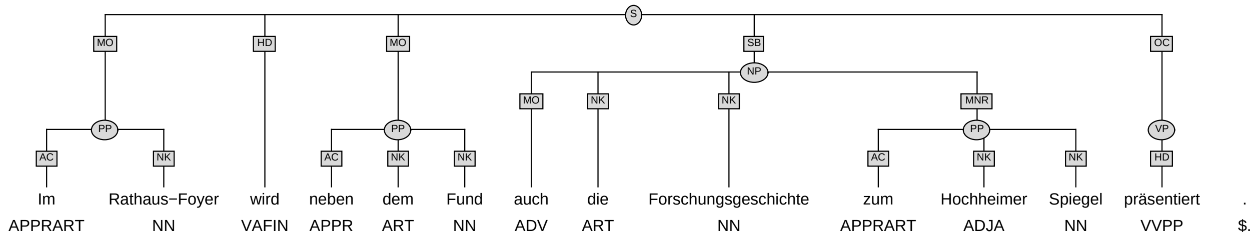

Figure 3: The NEGRA sentence from Figure 1 without crossing branches.

both noun phrases being explicitly annotated. The tree also contains several unary nodes, i.e. nodes with only one daughter, e.g. the verb phrases “fährt” (goes) and “heißt” (is called) or the street name “Lagerstraße”. The main ordering principle on the clausal level are the topological fields, long-distance relationships such as the relation between the noun phrase “eine Straße” (a street) and the extraposed relative clause “die heute noch Lagerstraße heißt” (which is still called Lagerstraße) are marked via functional labeling; *OA-MOD* specifies that this noun phrase modifies the accusative object *OA*.

4 Preprocessing the Treebanks

Most state-of-the-art parsers are based on context-free grammars. However, both treebanks do not completely adhere to the requirements of a CFG: Apart from NEGRA’s crossing branches, both treebanks contain sentences that consist of more than one tree. For all sentences, a virtual root node that groups all trees is inserted, and parenthetical trees are attached to the surrounding tree. The virtual root also ensures that the grammar has a single start symbol. In order to resolve NEGRA’s crossing branches, a script was used that is provided with the graphical annotation tool, which was used to annotate both treebanks[2]. The script isolates crossing constituents and attaches the non-head constituents higher up in the trees. After the conversion, the sentence in Figure 1 receives the tree structure shown in Figure 3. Both modifiers of the verb phrase have been reattached at the clause level in order to resolve the crossing branches. Unfortunately, the modified tree does not contain any information on the scope of the modifiers, which has previously been shown by the low attachment in the VP. Since crossing branches occur in approximately 30% of the sentences, we use a modified script to keep trace of the original phrase from which the constituent was moved. In this version, NEGRA+traces, the crossing modifier PPs in Figure 1 are assigned the function label *MO<VP* specifying that they are extracted from the verb phrase. Thus, the tree would be the same as in Figure 3, except for the function labels of the two reattached PPs.

5 Comparing Treebanks for Parsing

For the experiments, the statistical left-corner parser LoPar (Schmid 00) was used. Since the experiments are designed to show differences in parsing quality depending on the annotation decisions, the parser was

[2] Cf. `www.coli.uni-saarland.de/projects/sfb378/negra-corpus/annotate.html`

	NEGRA	NEGRA+traces	TüBa-D/Z
crossing brackets	1.07	n.a.	2.27
labeled recall	70.09%	n.a.	84.15%
labeled precision	67.82%	n.a.	87.94%
labeled F-score	68.94	n.a.	86.00
crossing brackets	1.04	1.03	1.93
function labeled recall	52.75%	49.03%	73.65%
function labeled precision	51.85%	50.49%	76.13%
function labeled F-score	52.30	49.75	74.87

Table 1: The results of comparing NEGRA and TüBa-D/Z.

used without (EM) training or lexicalization of the grammar.

For all the experiments reported here, only sentences with a length of maximally 40 words were used. These sentences were randomly split into 90% training data and 10% test data. The test data were kept fixed in order to enable error analysis. Since we did not want to have the results influenced by POS tagging errors, the parser was given the gold POS tags for the test sentences[3].

For each experiment, two different types of tests were performed: For one type, the data contained only syntactic constituents, i.e. the grammatical functions, which are shown as square boxes in the trees, were omitted. Thus, the rule describing the root node and its daughters in Figure 3 is represented as "S → PP VAFIN PP NP VP". These tests are reported below as "labeled precision" and "labeled recall". In the second type of tests, the syntactic categories were augmented by their grammatical function. Thus, the same rule extracted from the tree in Figure 3 now contains the grammatical function for each node: "S → PP-MO VAFIN-HD PP-MO NP-SB VP-OC". (Note that the root node is the only node in the tree that does not have a grammatical function.) These tests are reported below as "function labeled precision" and "function labeled recall".

The results of the experiments on the original treebanks after preprocessing are shown in Table 1. As reported above, NEGRA contains crossing branches in 30% of the sentences, which had to be resolved in preprocessing. Since in these sentences, attachment information is often not present, the experiment was repeated with the version of NEGRA that contains traces of moved constituents. This representation is closer to the TüBa-D/Z annotation which also contains such information for long distance relationships. The results show that the F-score for TüBa-D/Z is significantly higher than for NEGRA trees. In contrast, the number of crossing brackets is lower for NEGRA. The NEGRA results raise the question whether the low crossing brackets rate in NEGRA is only due to the low number of constituents in the trees. The percentage of nodes per words shows that while NEGRA trees contain on average 0.88 nodes per word, TüBa-D/Z trees contain 2.38 nodes per word. This leads to the question whether the deeper structures in TüBa-D/Z can be parsed reliably but may not be useful for further processing. Thus, a more detailed investigation is necessary.

This discussion leads to the question of how to evaluate the parsing results in a meaningful way. Generally, there are two possible evaluation methods that go beyond the calculation of precision and recall: an analysis of the different constituents and a task-based evaluation. The former approach can show for which categories there are differences between the annotation schemes. The latter approach tests the utility of the parser output for a specific task such as anaphora resolution or question answering. While this would provide valuable insight, the results would be difficult to generalize from the specific task. For the former approach, the equivalence of the different syntactic and functional categories must be presupposed. Such a comparison is only meaningful if both annotation schemes describe the same phenomena with the same categories. Unfortunately, for NEGRA and TüBa-D/Z, this assumption often does not hold. The most obvious area in which the two treebanks differ is the treatment of unary nodes: while TüBa-D/Z annotates such constituents, NEGRA does not allow unary branching. The differences in annotation are shown in Figure 4 for NEGRA and in Figure 5 for TüBa-D/Z. In these trees, it becomes obvious that the differences in annotation are widespread and do not only concern verbal phrases but also, for example, noun phrases,

[3] Thus, the results are slightly better than in setting where the POS tags are assigned automatically.

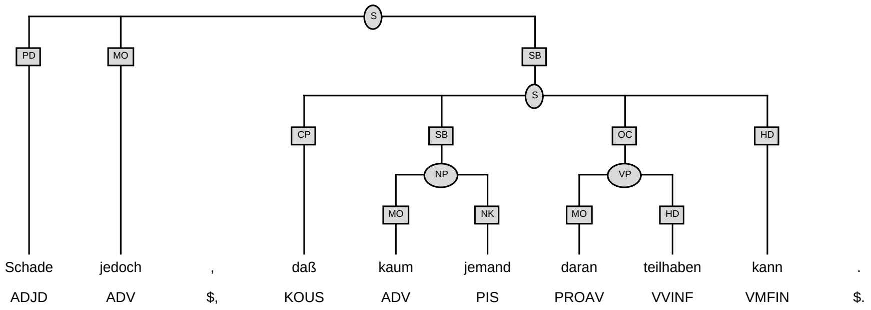

Figure 4: A sentence from NEGRA without the annotation of unary nodes.

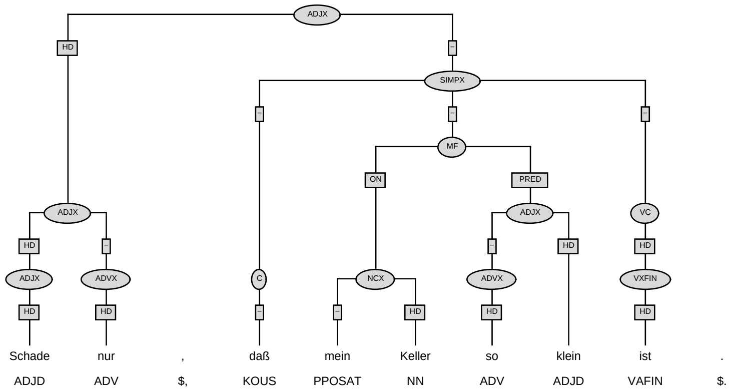

Figure 5: A sentence from TüBa-D/Z, in which unary nodes are annotated.

adverbial phrases, and prepositional phrases. Due to these great differences, a comparison of single constituents cannot be meaningful since one would compare, for example, all NPs in TüBa-D/Z to complexer NPs (with two words or more) in NEGRA.

Other differences concern the use of the POS tagset which are also reflected in phrase structure, e.g. stative passives, the attachment of relative clauses, and the treatment of comparative particles. For example, NEGRA treats comparative particles without a comparative semantic interpretation as prepositions, thus annotating such phrases as PPs. In TüBa-D/Z, in contrast, the presence of a comparative particle does not change the phrase type.

In the absence of more detailed methods of comparison, testing the effect of modifying individual annotation decisions gives insight into the factors that influence parsing results. As mentioned above, NEGRA and TüBa-D/Z differ in three major points (the fourth difference, crossing branches in NEGRA, is already addressed in preprocessing): flatter phrases and no unary nodes in NEGRA, and flatter structures on the clause level in TüBa-D/Z. In order to test the individual decisions, the opposite treebank is modified to also follow the respective decision. So in order to test the influence of not annotating unary nodes, all such nodes were removed from TüBa-D/Z while the other differences remained unchanged.

Consequently, the following modifications of the treebanks were executed:

- To test the influence of not annotating unary nodes (such as in NEGRA), all nodes with only one daughter were removed from TüBa-D/Z, preserving the grammatical functions. In the following section, this version will be named Tü_NU.

- To test the influence of NEGRA's flat phrase structure, phrases in TüBa-D/Z were flattened. This version will be named Tü_flat.

- In a third test, both modifications, the removal of unary nodes and the flattening of phrases were applied to TüBa-D/Z. The resulting tree for the sentence in Figure 2 is shown in Figure 6. This version will be named Tü_flat_NU.

- In order to test the influence of the flatter TüBa-D/Z structure on the clause level, topological fields were introduced into the NEGRA annotations. The topological fields were automatically extracted from the NEGRA corpus by the DFKI Saarbrücken. Since the NEGRA annotation in

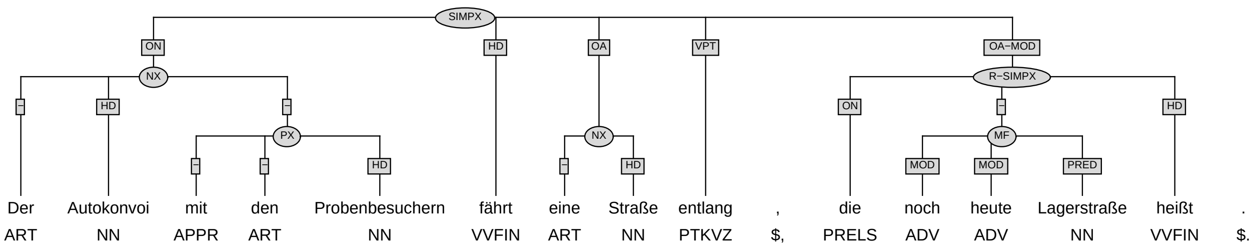

Figure 6: The sentence from Figure 2 in the flattened version without unary branches.

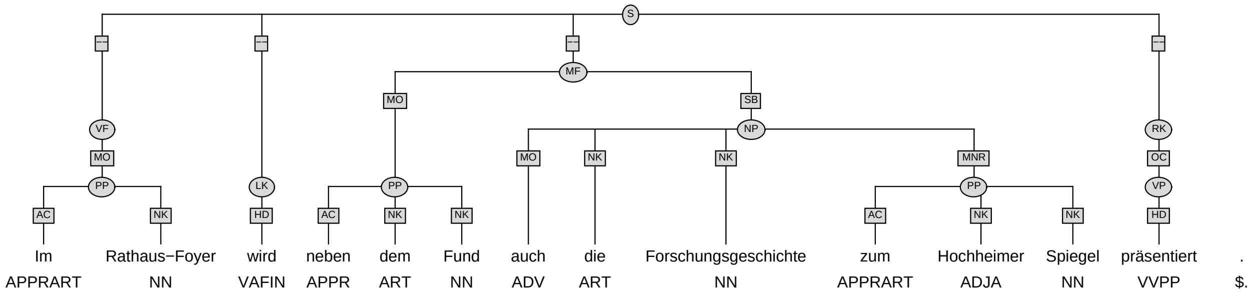

Figure 7: The sentence from Figure 1 with fields.

some cases does not contain enough information about the correct topological field, the conversion algorithm needs to use heuristics, which lead to a small number of errors in the field annotation.

The original annotation of NEGRA had to be modified when the topological fields were introduced. In many cases, the topological fields cross phrasal boundaries: These phrasal nodes were removed[4]. The resulting tree for the sentence in Figure 1 is shown in Figure 7. This version of NEGRA will be named NE_field.

The resulting modified treebanks were split into training and test data so that these sets contained the same sentences as the data sets for the baseline experiments. These data sets were then used for training and testing the parser on the modifications. The results of these experiments are shown in Table 2.

6 Discussion of the Results of the Comparison

Table 2 gives the results for the evaluation of the two types of tests: the upper half of the table gives resutls for parsing with syntactic node labels only and the lower half of the table gives results for parsing syntactic categories and grammatical functions. The results show that every transformation of the treebank annotations changes the results approximating those of the other treebank.

6.1 Modification of NEGRA

The modification of NEGRA, which **introduces topological fields** in order to flatten the clause structure, leads to an improved F-score but also to more crossing brackets. A first hypothesis would be that the improvement is due to the reliable recognition of the new field nodes. This hypothesis can be rejected by an evaluation of the parsing results for single syntactic categories. This evaluation shows that the introduction of topological fields gives high F-scores for the major fields, but it also improves both precision and recall for adverbial phrases, noun phrases, prepositional phrases, and almost all types of coordinated phrases, For adjectival phrases, precision improves from 55.95% to 64.46% - but at the same time, recall degrades from 56.38% to 50.97%. In contrast, the F-score for verb phrases deteriorates. This is probably due to the fact that only such verb phrases are annotated which do not cross field boundaries.

One reason for the improvement in the overall F-score is the change in the number of rules for a specific syntactic category. A look at the rules extracted from the training corpus shows a dramatic drop in numbers: for adjectival phrases, the number drops from more than 3900 rules containing AP to approximately3400 – even though new rules were added for the treatment of topological fields.

[4] We also tested a version in which the phrases were split into two to fit under the topological fields. However, this change resulted in lower precision and recall values.

	NEGRA	NE_field	TüBa-D/Z	Tü_NU	Tü_flat	Tü_flat_NU
crossing brackets	1.07	1.30	2.27	1.87	1.09	1.15
labeled recall	70.09%	75.21%	84.15%	77.41%	85.63%	77.43%
labeled precision	67.82%	77.17%	87.94%	81.52%	86.24%	76.44%
labeled F-score	68.94	76.18	86.00	79.41	85.93	76.93
sentences not parsed (%)	0.55%	0.05%	0.48%	1.91%	0.62%	2.26%
crossing brackets	1.04	1.21	1.93	2.17	1.07	1.29
function labeled recall	52.75%	69.85%	73.65%	62.11%	73.80%	53.63%
function labeled precision	51.85%	69.53%	76.13%	65.43%	74.66%	58.87%
function labeled F-score	52.30	69.19	74.87	63.73	74.23	56.13
sentences not parsed (%)	12.59%	2.17%	1.03%	9.98%	3.55%	18.87%
ratio nodes/words (in treebank)	0.88	1.38	2.38	1.33	2.00	1.06

Table 2: The results of comparing the modified versions of NEGRA and TüBa-D/Z.

6.2 Modification of TüBa-D/Z

Each modification of TüBa-D/Z results in a loss in F-score, but also in an improvement concerning crossing brackets. While flattening phrase structure only leads to minor changes, **deleting unary nodes** has a detrimental effect: the F-score drops from 86.00 to 79.41 when parsing syntactic constituents, and from 74.83 to 63.73 when parsing syntactic constituents including grammatical functions. Especially when parsing grammatical functions, deleting unary nodes leads to an increase of sentences that could not be parsed by a factor of almost 10. These sentences would have required additional rules not present in the training sentences. This leads to the question whether the deterioration is only due to the high number of sentences which were assigned no parse. However, an evaluation of only those sentences that did receive a parse shows only slightly better results in recall (obviously, precision remains the same): 68.18% for parsed sentences as compared to 62.11% for all sentences. This result, however, may also be caused by missing rules, which is corroborated by a look at the rules extracted from the test sentences: Approximately 24.0% of the rules needed for correctly parsing the test sentences in the modification without unary nodes are not present in the training set, as compared to 18.2% in the original version of the TüBa-D/Z treebank.

A closer look at the different constituents shows that the syntactic categories that are affected most by the **deletion of unary nodes** are noun phrases, finite verb phrases, adjectival phrases, adverbial phrases, and infinitival verb phrases. All those categories suffer losses in the F-score between 1.81% (for infinitival verb phrases) and 57.28% (for adverbial phrases). Since both precision and recall are similarly affected, this means that the parser does not only annotate spurious phrases but also misses phrases which should be annotated.

Flattening phrases in TüBa-D/Z has a negative effect on precision but it causes a slight increase in recall. The latter effect is a consequence of the bias of the PCFG parser, which prefers small trees. A comparison of the average number of nodes per word in a sentence shows that for all models, the parsed trees contain significantly fewer nodes than the gold standard trees. For the original TüBa-D/Z grammar including grammatical functions, the parsed tree contains 54.6% of the nodes in the gold standard; in the flattened version, the ratio is 58.6% (and for NEGRA, it is 62.5%).

The category that profits most from this modification is the category of named entities (*EN-ADD*). This is not surprising considering the fact that this node type does not serve a syntactic function, it is inserted above the syntactic category, which spans the named entity (cf. e.g. the named entity “Lagerstraße” in Figure 2). Flattening the structure often deletes the internal node and consequently allows the parser to base the annotation of named entities on more information than just a noun phrase node. This result is even more pronounced when also unary nodes are deleted. Other syntactic categories that profit from a flattening of the trees are prepositional phrases and relative clauses.

The **combination of both modifications** in TüBa-D/Z, flat phrase structure and deleted unary nodes, leads to a dramatic loss in the F-score for functional parsing as compared to the experiment in which only the unary nodes were deleted. A look at the unlabeled F-scores shows that this loss is not only due to incorrect labels for constituents, it also affects the recognition of phrase boundaries: the unlabeled F-score degrades from 91.34 for the original version of Tüba-

D/Z, to 81.06 for the version without unary nodes, and to 71.65 for the combination of both modifications.

7 Conclusion and Future Work

We have presented a method for comparing different annotation schemes and their influence on PCFG parsing. It is impossible to compare the performance of a parser on single syntactic categories since even rather similar annotation schemes apply different definitions for different phrase types. As a consequence, the comparison must be based on modifications within one annotation scheme to make it more similar to the other. The experiments presented here show that annotating unary nodes and structured phrases improve parsing results. On the clause level, however, a flatter structure incorporating topological fields is helpful for German.

The experiments presented here were conducted with a standard PCFG parser. The next logical step is to extend the comparison to different probabilistic parsers with different probability models and different biases. The (Charniak 00) parser or in the (Klein & Manning 03) parser use extensions of the probability model which were very successful for English. It is, however, unclear what the effect of these extensions is on German data.

Another area to be explored is lexicalization. Here, the picture is also unclear: Studies on the Penn treebank show that parsing results improve with lexicalized trees (cf. e.g. (Collins 97; Charniak 00)). The results on German (Dubey & Keller 03), however, show a detrimental effect of lexicalization for the NEGRA data. Thus, a comparison of treebank annotation schemes based on lexicalization only makes sense if a method of lexicalization can be found for both annotation schemes that does not overly decrease performance.

Another unexplored area for the two treebanks used here is the difference in grammatical functions. A comparison of grammatical functions, however, cannot be performed on the basis of a modification from one set to the other since there is no straightforward conversion from one set of grammatical functions to the other. For such a comparison, task-based evaluations of the parser trained on the two treebanks will be necessary.

Acknowledgments

We are grateful to the DFKI Saarbrücken for making their topological field annotation for the NEGRA treebank available to us. We would also like to thank Wolfgang Maier, Julia Trushkina, Holger Wunsch, and Tylman Ule for providing scripts for the conversion and evaluation of the data, and Erhard Hinrichs and Tylman Ule for many fruitful discussions.

References

(Bosco *et al.* 00) Cristina Bosco, Vincenzo Lombardo, D. Vassallo, and Leonardo Lesmo. Building a treebank for Italian: a data-driven annotation scheme. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation, LREC-2000*, Athens, Greece, 2000.

(Charniak 00) Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics and the 6th Conference on Applied Natural Language Processing, ANLP/NAACL'2000*, pages 132–139, Seattle, WA, 2000.

(Charniak 01) Eugene Charniak. Immediate head parsing for language models. In *Proceedings of the 39th Annual Meeting of the ACL and the 10th Conference of the European Chapter of the ACL, ACL/EACL 2001*, pages 116–123, Toulouse, France, 2001.

(Collins 97) Michael Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the ACL (jointly with the 8th Conference of the EACL)*, Madrid, Spain, 1997.

(Drach 37) Erich Drach. *Grundgedanken der Deutschen Satzlehre*. Diesterweg, Frankfurt/M., 1937.

(Dubey & Keller 03) Amit Dubey and Frank Keller. Probabilistic parsing for German using sister-head dependencies. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 96–103, Sapporo, Japan, 2003.

(Höhle 86) Tilman Höhle. Der Begriff "Mittelfeld", Anmerkungen über die Theorie der topologischen Felder. In *Akten des Siebten Internationalen Germanistenkongresses 1985*, pages 329–340, Göttingen, Germany, 1986.

(Klein & Manning 03) Dan Klein and Christopher Manning. Accurate unlexicalized parsing. In *Proceedings of ACL-2003*, pages 423–430, Sapporo, Japan, 2003.

(Marcus *et al.* 94) Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the ARPA Human Language Technology Workshop, HLT 94*, Plainsboro, NJ, 1994.

(Montegmagni *et al.* 00) S. Montegmagni, F. Barsotti, M. Battista, N. Calzolari, O. Corazzari, A. Zampolli, F. Fanciulli, M. Massetani, R. Raffaelli, R. Basili, M. T. Pazienza, D. Saracino, F. Zanzotto, N. Mana, F. Pianesi, and R. Delmonte. The Italian syntactic-semantic treebank: Architecture, annotation, tools and evaluation. In *Proceedings of the Workshop on Linguistically Interpreted Corpora LINC-2000*, pages 18–27, Luxembourg, 2000.

(Sampson 93) Geoffrey Sampson. The SUSANNE corpus. *ICAME Journal*, 17:125 – 127, 1993.

(Schmid 00) Helmut Schmid. LoPar: Design and implementation. Technical report, Universität Stuttgart, 2000.

(Skut *et al.* 97) Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*, Washington, D.C., 1997.

(Telljohann *et al.* 04) Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. The TüBa-D/Z treebank: Annotating German with a context-free backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2229–2235, Lisbon, Portugal, 2004.

(Thielen & Schiller 94) Christine Thielen and Anne Schiller. Ein kleines und erweitertes Tagset fürs Deutsche. In Helmut Feldweg and Erhard Hinrichs, editors, *Lexikon & Text*, pages 215–226. Niemeyer, Tübingen, 1994.