How to Compare Treebanks

Sandra Kübler, Wolfgang Maier, Ines Rehbein, Yannick Versley

Indiana University, Universität Tübingen, Dublin City University, Universität Tübingen
skuebler@indiana.edu, wo.maier@uni-tuebingen.de, irehbein@computing.dcu.ie, versley@sfs.uni-tuebingen.de

Abstract

Recent years have seen an increasing interest in developing standards for linguistic annotation, with a focus on the interoperability of the resources. This effort, however, requires a profound knowledge of the advantages and disadvantages of linguistic annotation schemes in order to avoid importing the flaws and weaknesses of existing encoding schemes into the new standards. This paper addresses the question how to compare syntactically annotated corpora and gain insights into the usefulness of specific design decisions. We present an exhaustive evaluation of two German treebanks with crucially different encoding schemes. We evaluate three different parsers trained on the two treebanks and compare results using EVALB, the Leaf-Ancestor metric, and a dependency-based evaluation. Furthermore, we present TePaCoC, a new testsuite for the evaluation of parsers on complex German grammatical constructions. The testsuite provides a well thought-out error classification, which enables us to compare parser output for parsers trained on treebanks with different encoding schemes and provides interesting insights into the impact of treebank annotation schemes on specific constructions like PP attachment or non-constituent coordination.

1. Introduction

Interoperability has become an important issue in the development of language resources, as recent events such as the Workshop on Multilingual Language Resources and Interoperability at ACL 2006 or the Workshop "Toward the Interoperability of Language Resources" at the LSA Summer Institute 2007 prove. One aspect of interoperability is the adaptation of existing syntactic annotation schemes for new languages. This strategy has been used in the annotation of the Penn Arabic Treebank (Bies and Maamouri, 2003), or the Penn Korean Treebank (Han et al., 2001), which use adaptations of the Penn English Treebank annotation scheme (Bies et al., 1995). However, such adaptations are risky as long as we do not know how the decisions in the annotation scheme influence parser performance and parser evaluation. Previous work on determining the influence of the annotation scheme has concentrated on German because of the ideal situation with regard to treebank resources: There exist several treebanks for written German, which are based on two different annotation schemes: The NEGRA/TIGER annotation scheme (Brants et al., 2002) uses crossing branches for long-distance relationships and a flat annotation of phrase and clause structure while the TüBa-D/Z annotation scheme (Telljohann et al., 2005) favors hierarchical structures, uses topological fields, and annotates long-distance relationships by special functional labels. First results in the comparison of these two treebanks (Kübler, 2005; Maier, 2006) show that the differences in precision, recall, and F-score between parsers trained and tested on these two treebanks are in the area of 20 percent points. Kübler (2005) and Maier (2006) attribute the differences to the learnability of the grammars by statistical parsers; they argue that the TIGER grammar has a narrower coverage because the flat tree structure results in many rules with long right sides. As an additional problem of the TIGER annotation scheme, the strategy generally used to resolve the crossing branches is mentioned: Crossing non-head constituents are passed up in the tree until they are attached to a node where they do not cross anymore. Information about modifier scope gets lost during this transformation. Rehbein and van Genabith (2007), in contrast, argue that the higher F-scores for the TüBa-D/Z do not reflect better quality in the parser output but are due to the higher ratio of non-terminal vs. terminal nodes in the TüBa-D/Z, which results in an overall higher number of brackets in the trees. Given that PARSEVAL F-scores are computed relative to the number of brackets in the tree, a bracket mismatch in TüBa-D/Z is considered less severe than in TIGER.

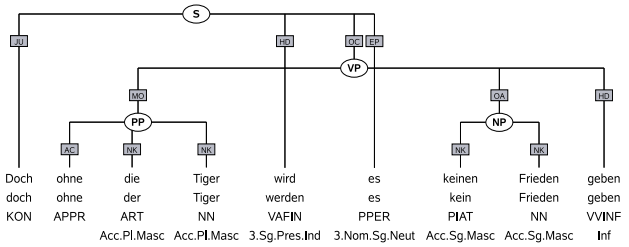
The results of these studies are contradictory in many of their findings, so a more fine-grained evaluation is urgently needed. In the investigation at hand, we extend the standard PARSEVAL evaluation, EVALB, by applying the Leaf-Ancestor metric (Sampson and Babarczy, 2003) and a dependency-based evaluation, as well as a manual evaluation of a carefully selected set of sentences displaying grammatical phenomena which are extremely difficult to parse. The results of this investigation provide further evidence that the design of the syntactic annotation scheme has a significant influence on parser performance as well as on the evaluation.

The paper is structured as follows: In Section 2., we present a testsuite for the manual evaluation of parser performance on specific grammatical constructions and give an overview over the main properties of the two German treebanks used in our parsing experiments. Section 3. describes the experimental setup, and Section 4. discusses the results. The last section concludes.

2. Testing Parser Performance on Complex Grammatical Constructions (TePaCoC)

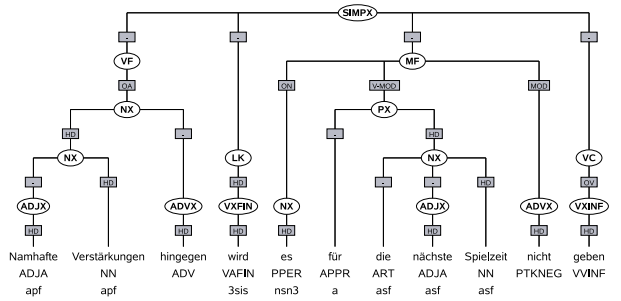
In order to test the performance of different parsers and compare the outcome for the two treebank annotation schemes, we created TePaCoC, a testsuite consisting of 200 sentences (100 from each treebank described below) with grammatical constructions which pose a challenge for a statistical parser. We concentrated on the following phenomena:

1. Extraposed Relative Clauses (ERC)
2. Forward Conjunction Reduction (FCR)



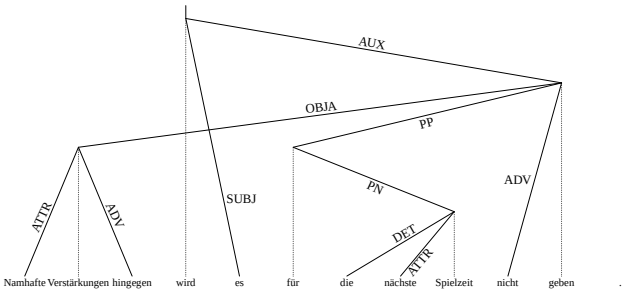
"But without the Tigers there will be no peace"

Figure 1: TIGER treebank tree



"However, there won't be considerable reinforcements for the next playing time

Figure 2: TüBa-D/Z treebank tree



"However, there won't be considerable reinforcements for the next playing time"

Figure 3: Dependencies for the TüBa-D/Z sentence

3. Coordination of Unlike Constituents (CUC)
4. Noun PP Attachment (PPN)
5. Verb PP Attachment (PPV)
6. Subject Gap with Finite/Fronted Verbs (SGF)

2.1. Data sources: TIGER and TüBa-D/Z

The data in the testsuite is taken from two different sources: the TIGER treebank (Release 2) (Brants et al., 2002) and the TüBa-D/Z (Release 3) (Telljohann et al., 2005). Both treebanks contain German newspaper text and are annotated with phrase structure and dependency (functional) information. Both treebanks use the same POS tag set (STTS) (Schiller et al., 1995), but the number of category labels and grammatical function labels varies dramatically. Table 1 gives an overview over some features of the two treebanks. The most important differences between the two treebanks are: (1) the annotation in TIGER is rather flat compared to the more hierarchical annotation in TüBa-D/Z, (2) TIGER does not annotate unary branching, (3) TüBa-D/Z annotates topological fields, and (4) long distance relationships are expressed via crossing branches in TIGER while in TüBa-D/Z, the same phenomenon is expressed with the help of special grammatical function labels (e.g. OA-MOD for a constituent that modifies the direct object OA).

Figures 1 and 2 illustrate some of the differences between the two treebanks. In the TIGER example (Figure 1), the PP *ohne die Tiger* (without the Tigers) modifies the verb *geben* (to give), and is therefore attached to the VP node. This results in crossing branches. In the TüBa-D/Z example (Figure 2), the functional label VMOD is used to express the same relationship between the PP *für die nächste Spielzeit* (for the next playing time) and the verb. The examples also show the flat annotation of PPs in TIGER (Figure 1), compared to the the more hierarchical annotation in TüBa-D/Z (Figure 2). The annotation of unary nodes as well as the additional level of topological fields in TüBa-D/Z result in a much higher ratio of non-terminal versus terminal nodes than in the TIGER treebank (Table 1).

There are also considerable differences with regard to the use of grammatical functions in the two treebanks. One example is the expletive *es* (it), which in the TIGER example (Figure 1) is assigned the grammatical function label EP (expletive), while in TüBa-D/Z (Figure 2) the expletive *es* is annotated as a nominative object (= subject).

All sentences, parser output and treebank sentences, were converted to dependencies. The conversion aimed at finding dependency representations for both treebanks that are as similar to each other as possible. Complete identity is impossible because the treebanks contain different levels of distinction for different phenomena. The conversion is based on the original formats of the treebanks. The target dependency format was defined based on the dependency grammar by Foth (2003). The dependency version of the TüBa-D/Z tree from Figure 2 is shown in Figure 3.

	# sent.	avg. sent. length	cat. node labels	GF labels	non-term. /term. nodes
TIGER	50474	17.46	25	44	0.47
TüBa-D/Z	27125	17.60	26	40	1.20

Table 1: Some features of TIGER and TüBa-D/Z

2.2. Creating TePaCoC

For each of the grammatical phenomena listed above, we selected 20 sentences from TIGER and from TüBa-D/Z with a sentence length ≤ 40, containing the particular construction. This results in a testset of 200 sentences, 100 from each treebank, which allows us to assess the impact of specific treebank design decisions on parser performance.

Both treebank annotation schemes employ different means to encode the same phenomena, which makes a direct com-

parison of the parser output for the two treebanks non-trivial. Therefore, we developed an error classification system which enables us to judge the quality of the parser output trees across different treebanks.

Table 2 shows the error classification for the case of Extraposed Relative Clauses (ERC). In TIGER, the grammatical function label RC carries the information that the clause is a relative clause (Figure 4), while in TüBa-D/Z, the same information is encoded in the categorial node label R-SIMPX (Figure 5). Therefore, (A) in Table 2 corresponds to a function label error in TIGER and to a categorial node label error in TüBa-D/Z. The relationship between the relative clause and its head noun is expressed via attachment in TIGER and by the use of a grammatical function label in TüBa-D/Z. So (B) is caused by a wrong attachment decision for a parser trained on TIGER and by a grammatical function label error for a parser trained on TüBa-D/Z. For (C), the parser fails to identify the relative clause at all. In TüBa-D/Z, this is usually caused by a POS tagging error, where the parser fails to assign the correct POS tag to the relative pronoun. (D) applies to both annotation schemes: here, the main components of the clause have been identified correctly but the phrase boundaries are slightly wrong. The use of the error classification scheme guarantees the reliability and consistency of the manual evaluation.

3. Experimental Setup

In order to cover all possible reasons for the differences in the treebanks, we evaluate on three levels: First, we use EVALB (an implementation of the PARSEVAL metric) and the Leaf-Ancestor metric (Sampson and Babarczy, 2003) to evaluate the constituents, then we convert the two annotation schemes into the same dependency format and evaluate the dependencies. Since we use the same (or as similar as possible) set of dependencies, the conversion should abstract away from differences in the two treebanks. In a third step, we perform a manual evaluation of the phenomena covered in TePaCoC.

EVALB is well known to excessively punish attachment error over several levels in the tree (Kübler and Telljohann, 2002) and to give better results for annotation schemes with a deep hierarchical structure. For this reason, we added the Leaf-Ancestor metric, which has been shown to be less biased towards annotation schemes with a higher ratio of non-terminal vs. terminal nodes (Rehbein and van Genabith, 2007). However, Leaf-Ancestor is still sensitive to the number of brackets in the trees. Another way of leveling out differences between constituent annotations is the conversion of constituent structure to dependencies. In this case, each word has exactly one dependency that relates it to its head so that all annotations are reduced to the most important attachment information. This evaluation strategy goes back to Lin (1995; 1998).

For the experiments, we removed the TePaCoC sentences from the treebanks and divided the remaining sentences into a training set of 25 005 sentences and a testset of 2 000 sentences (the remaining TIGER sentences were ignored). Then we trained the unlexicalized parsers BitPar (Schmid, 2004) and LoPar (Schmid, 2000), and the Stanford parser (Klein and Manning, 2003) in its lexicalized and markovized form[1] on the training set and tested them on the 2 000 test sentences as well as on the 200 TePaCoC sentences.

	TIGER			TüBa-D/Z		
	Bit	**Lop**	**Stan**	**Bit**	**Lop**	**Stan**
EVALB	74.0	75.2	77.3	83.4	84.6	88.5
LA	90.9	91.3	92.4	91.5	91.8	93.6

Table 3: EVALB and LA scores (2 000 sentences)

Before extracting the grammars, we resolved the crossing branches in TIGER and, where grammatical function labels such as *subject* or *accusative object* were directly attached to the terminal node, we inserted an additional unary node to prevent blowing up the POS tagset for the TIGER grammar. In all experiments, we used raw text as parser input and let the parsers assign the POS tags.

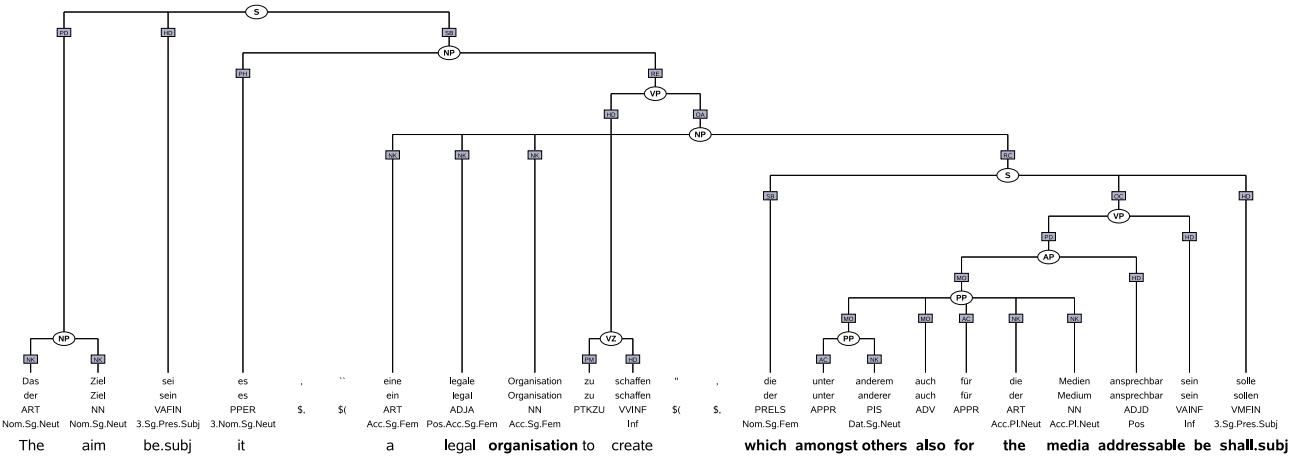
For the dependency-based evaluation, we converted the phrase-structure annotations into dependencies according to the German Dependency Grammar of Foth (2003). For this task, we used pre-existing dependency converters for TIGER-style trees (Daum et al., 2004) and for TüBa-D/Z-style trees (Versley, 2005). While imperfections in the conversion exist and may slightly lower the results especially when comparing TüBa-D/Z parses with a TIGER gold standard or vice versa, comparing the accuracy of common grammatical functions usually provides a robust quality estimate for parses.

4. Results

4.1. Constituent Evaluation

Table 3 shows EVALB and LA scores for the 2 000 sentence testsets. There is a wide gap between EVALB results for the TIGER and the TüBa-D/Z model while LA scores for both treebanks are much closer. This is due to the fact that EVALB has a strong bias towards annotation schemes with a high ratio of nonterminal vs. terminal nodes as in the TüBa-D/Z (Rehbein and van Genabith, 2007). Additionally, there is a clear improvement from BitPar to LoPar to the Stanford parser for both treebanks, which is consistent for both constituency-based evaluation metrics. The differences between BitPar and LoPar are rather surprising since both parsers are based on the same principles. The difference may be due to the internal translation of the grammar into CNF in BitPar (Schmid, 2004). The Stanford parser obviously profits from the combination of lexicalization and markovization.

Table 4 shows the results for the TePaCoC sentences. In comparison to the testset, most scores are considerably lower, which shows that the TePaCoC sample is, on average, more difficult to parse. Again, the general trend is an improvement from BitPar to LoPar to the Stanford parser, which is consistent with the results for the 2 000 sentence testsets. And similar to the testsets, the results for TüBa-D/Z are higher than for TIGER. Section 4.2. discusses the different behavior of the two evaluation metrics in detail.



"The goal should be to create a legal organisation which, amongst others, is also accessible for the media."

Figure 4: Annotation of Extraposed Relative Clauses in TIGER

Error description		TIGER	TüBa-D/Z
(A)	Clause not recognized as rel. clause	Grammatical function incorrect	SIMPX label instead of R-SIMPX
(B)	Head noun incorrect	Attachment error	Grammatical function incorrect
(C)	Clause not recognized	Clause not recognized	Clause not recognized
(D)	Clause boundaries not correct	Span error	Span error

Table 2: Error classification for Extraposed Relative Clauses

		TIGER			TüBa-D/Z		
		Bit	**Lop**	**Stan**	**Bit**	**Lop**	**Stan**
EVALB	**ERC**	71.7	73.0	76.1	80.6	82.8	82.8
	FCR	76.6	77.7	81.3	84.0	85.2	86.7
	PPN	71.2	73.9	83.6	86.2	87.4	89.2
	PPV	71.9	76.5	78.7	84.3	85.0	91.9
	CUC	55.9	56.5	63.4	78.4	73.6	76.6
	SGF	73.3	74.1	78.6	73.6	76.6	78.4
	ALL	**69.64**	**71.07**	**75.82**	**81.20**	**83.51**	**84.86**
LA	**ERC**	85.3	86.1	84.8	89.3	89.8	91.0
	FCR	91.2	89.0	91.0	92.0	93.4	88.7
	PPN	87.1	88.7	91.0	94.2	94.3	94.4
	PPV	88.4	88.9	86.4	91.3	90.5	94.7
	CUC	78.0	78.4	78.3	82.2	85.5	84.9
	SGF	89.1	89.7	87.5	90.9	94.4	88.5
	ALL	**86.26**	**86.42**	**86.09**	**89.42**	**91.13**	**89.84**

Table 4: EVALB (labeled) bracketing and LA scores

(S She saw (NP the dog) (PP with the telescope))
(S She saw (NP the dog (PP with the telescope)))

2 out of 3 brackets correct → **66.7%** *labelled f-Score*

Figure 6: EVALB result for the TIGER-encoded example

4.2. Discussion: LA versus evalb

Table 3 shows a great gap between EVALB results for TIGER and TüBa-D/Z, while LA scores for the two treebanks are quite close. We will illustrate the differences between the two evaluation measures with the help of the English example sentence below:

[1]The parser was trained using the following parameters for markovization: `hMarkov=1, vMarkov=2`.

(S (VF (NP She)) (LK (VP saw)) (MF (NP the dog**)** (PP with (NP the telescope)))
(S (VF (NP She)) (LK (VP saw)) (MF (NP the dog (PP with (NP the telescope))**))**
7 out of 8 brackets correct → **87.5%** *labelled f-Score*

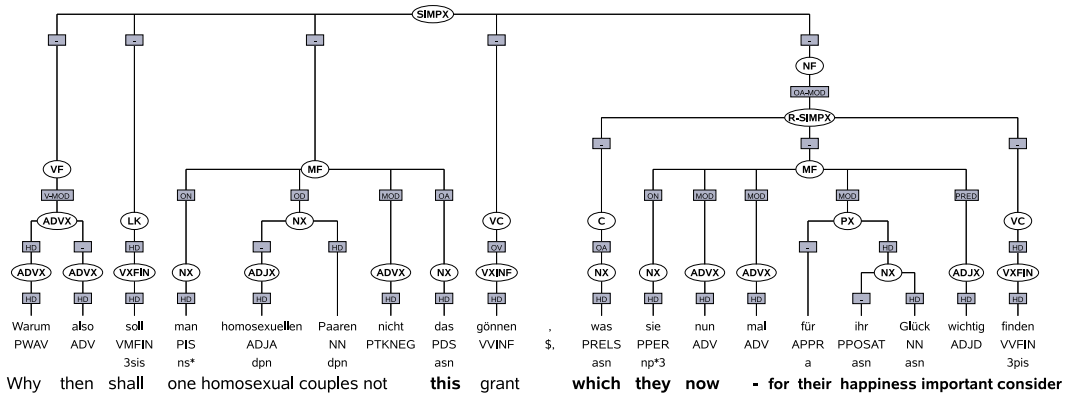
Figure 7: EVALB result for the TüBa-D/Z -encoded example

She saw the dog with the telescope. (1)

Figure 6 shows example (1) in the TIGER encoding scheme. The first representation gives the tree for the intended meaning of the sentence, while in the second tree, the PP *with the telescope* is falsely attached to the noun *dog*. EVALB evaluates the trees by counting the matching brackets in both trees. For the two TIGER-encoded sentences, EVALB gives the result shown in Figure 6.

If we take the same sentence and annotate it according to the TüBa-D/Z encoding scheme, the results are slightly different, as shown in Figure 7.

EVALB measures parser quality by counting matching brackets in the gold tree and the parser output. For the more hierarchical annotation scheme of the TüBa-D/Z where the deep annotation results in a higher number of brackets for each tree, the effect of a mismatching bracket is less severe than for TIGER. This shows that the PARSEVAL metric is highly biased towards annotation schemes with a high ratio of nonterminal vs. terminal nodes.



"So why shouldn't homosexual couples be granted what they think is important to their happiness."

Figure 5: Annotation of Extraposed Relative Clauses in TüBa-D/Z

In contrast to this, LA is a string-based similarity measure which extracts the path from the root node to each terminal node in the tree and calculates the cost of transforming the parser output tree into the gold tree. Each path consists of the sequence of node labels between the terminal node and the root node, and the similarity of two paths is calculated with the help of the Levenshtein distance (Levenshtein, 1966). In order to account for linguistically more or less severe errors, LA charges a higher cost for the substitution of two unrelated node labels, while the replacement of two related labels such as a VP and a VZ (infinitive with *zu* in TIGER) or a VXFIN and a VXINF (finite vs. infinite verb phrase in TüBa-D/Z) is rather cheap.

Consider the TIGER example sentence in Figure 6. The LA metric would extract the paths listed in the upper part of Table 5 for each terminal node in the trees. POS tags are not represented in the paths. The principles for the insertion of phrase boundaries, expressed through square brackets, are described in Sampson and Babarczy (2003).

TIGER		**gold path**		**parser output**
1.000	She	[S	:	[S
1.000	sees	S	:	S
1.000	the	[NP S	:	[NP S
0.800	man	NP] S	:	NP S
0.857	with	[PP S	:	[PP NP S
0.800	the	NP S	:	PP NP S
0.857	telescope	PP] S	:	PP NP] S
0.902	*average score for TIGER*			

TüBa-D/Z		**gold path**		**parser output**
1.000	She	NP VF] [S	:	NP VF] [S
1.000	sees	VP [LK] S	:	VP [LK] S
1.000	the	NP [MF S	:	NP [MF S
0.857	man	NP] MF S	:	NP MF S
0.889	with	[PP MF S	:	NP MF S
0.909	the	[NP PP MF S	:	[NP PP NP MF S
0.909	telescope	NP PP MF S]	:	NP PP NP MF S]
0.938	*average score* for TüBa-D/Z			

Table 5: LA paths and scores for the example sentence

LA assigns a score to each terminal node in the tree. Identical paths are assigned a score of 1, and the score for the whole sentence is the average over all scores for this particular tree. For example (1) in the TIGER encoding, the LA score is 0.902.

The lower part of Table 5 shows LA results for the same sentence in TüBa-D/Z encoding. While there was a gap of around 20% between EVALB results for example (1), LA results for the TüBa-D/Z encoded sentence are only around 4% better than for TIGER. The better LA results for the TüBa-D/Z encoded sentence are due to the fact that while the same three terminals are affected by the error as for TIGER, due to the more hierarchical annotation and the extra layer of topological fields, the paths in the TüBa-D/Z annotation scheme are longer than in TIGER. Therefore the edit cost for inserting or deleting one symbol in the path, which is computed relative to path length, is lower for the TüBa-D/Z trees. This shows that the LA metric is also biased towards the TüBa-D/Z, but not to such a great extent as the PARSEVAL metric.

4.3. Dependency Evaluation

The strong bias in the two constituency-based metrics caused us to amend the evaluation results by adding a dependency-based evaluation. As described above, for the dependency evaluation, both treebank sentences and parsed sentences were converted to dependency representations that were as similar between the two treebanks as possible. Foth's (2003) manual distinguishes 34 dependency relations, with distinctions between different verb arguments (5 relations), different kinds of clausal subordinations (infinitive, dependent object/adjunct clause, full sentence, and relative clauses), and several adjunct relations, which mostly depend on the part of speech of the adjunct; We followed Versley (2005) in conflating the labels for prepositional adjuncts and arguments to verbs, since this distinction is not consistent across different annotation schemes. In contrast to approaches that are oriented towards shallow semantics, such as the Tiger Dependency Treebank (Forst, 2003), Foth's dependency grammar only considers syntactic rela-

	TIGER			TüBa-D/Z		
	Bit	**Lop**	**Stan**	**Bit**	**Lop**	**Stan**
LAS	78.8	80.5	81.6	71.3	72.8	75.9
UAS	83.0	84.5	85.6	81.7	83.4	86.8

Table 6: Labeled/unlabeled dependency accuracy for the 2 000 test sentences

	TIGER			TüBa-D/Z		
	Bit	**Lop**	**Stan**	**Bit**	**Lop**	**Stan**
SUBJ	80.2	81.1	78.7	74.6	75.3	76.1
OBJA	55.6	58.4	59.5	42.4	45.8	52.9
OBJD	11.6	11.5	14.1	12.9	13.3	13.1
PP	71.1	72.2	78.2	68.1	69.1	75.6
cl-sub	57.0	58.2	60.9	45.8	47.5	52.1

Table 7: Dependency F-measure for the 2 000 test sentences: nominal verb arguments (subjects and accusative/dative objects), PP attachment and clause subordination (including infinitive and relative clauses as well as adjunct and argument subordinated clauses and argument full clauses)

tions and does not make reference to lexical semantic properties such as subject- or object control, and has exactly one head word for each dependent word, which also means that the influence of wrong attachments on the attachment score is very predictable.

Table 6 shows the results of the evaluation of the 2 000 sentence testsets after their conversion to dependencies. For both TIGER and TüBa-D/Z, we see again the improvement from BitPar to LoPar to the Stanford parser, with the Stanford parser reaching the highest unlabeled accuracy on TüBa-D/Z, and the highest labeled accuracy on TIGER. A comparison of TIGER and TüBa-D/Z results generally shows that the labelled accuracy for TIGER is higher than for TüBa-D/Z, thus contradicting the results of the constituent evaluation. If we look at the results on the testsuite (Table 8), we find that it generally confirms the tendencies about improvement along the BitPar-LoPar-Stanford parser axis, and a higher labeled accuracy for the TIGER parses. However, while labeled accuracy sees an improvement for the Stanford parser on TIGER, there is a decrease in labeled accuracy on the TüBa-D/Z part of the testsuite, where the Stanford parser shows improvement on the PP-attachment subset, but struggles with all the coordination cases, especially the subject-gap-fronted (-2.6) and coordination of unlikes (-2.1). This is consistent with the assumption that topological fields, while generally being helpful, prevent the realization of benefits from lexicalization on the clause level; furthermore, this shows that the question of which kind of information is suitable to support parser performance is highly dependent on the data structures in the training data.

4.4. Manual Evaluation of TePaCoC Phenomena

The manual evaluation of the constituent parses is intended to complement the automatic evaluation of the complete trees of the testsuite. This is necessary to ensure that the previous evaluations of the testsuite sentences did not concentrate on other phenomena or were contorted by the conversion to dependencies.

	TIGER			TüBa-D/Z		
	Bit	**Lop**	**Stan**	**Bit**	**Lop**	**Stan**
LAS ERC	76.2	76.0	77.4	71.6	71.8	71.1
FCR	79.5	74.4	81.8	78.5	81.0	79.3
PPN	76.8	79.7	87.0	75.5	76.1	76.1
PPV	73.6	80.9	79.2	65.8	67.9	71.5
CUC	65.2	67.0	70.7	57.5	63.0	60.9
SGF	76.1	77.2	79.3	74.0	77.7	75.1
ALL	**73.3**	**73.9**	**76.8**	**69.3**	**72.7**	**70.3**
UAS ERC	81.1	80.8	82.0	79.1	80.5	79.1
FCR	82.7	77.8	85.6	85.4	88.2	88.7
PPN	84.2	86.4	89.3	84.8	85.3	85.9
PPV	78.1	86.0	86.0	81.3	82.9	88.6
CUC	69.7	71.5	74.7	66.1	72.0	73.6
SGF	81.7	82.5	83.6	82.8	86.2	85.4
ALL	**78.1**	**78.7**	**81.0**	**78.3**	**81.9**	**81.7**

Table 8: Labeled/unlabeled dependency accuracy for the testsuite

The first result of this manual evaluation concerns the textual differences between the two treebanks. Since they are based on two different newspapers, there is the possibility that one text is more difficult to parse than the other. The manual evaluation of the testsuite, which contains sentences from both treebanks for each phenomenon, shows that there are no significant differences between the two sets of sentences in the testsuite.

Table 9 shows the results for human evaluation for the different phenomena in TePaCoC in terms of the number of sentences (or rather occurrences since some sentences contain more than one occurrence of the phenomenon) in which the respective phenomenon was parsed correctly. For Extraposed Relative Clauses, Forward Conjunction Reduction, and Subject Gaps with fronted finite verbs, all three parsers yield better results in the TIGER model. In the case of the relative clauses, the direct attachment of the relative pronoun to the relative clause node in TIGER makes it easier for the parser to recognize relative clauses. Additionally, since extraposed relative clauses were originally attached via crossing branches, which were then resolved, they are attached on the VP or on the clause level, making it easier for the parser to attach them correctly. In TüBa-D/Z, the relative pronoun projects to an NP, which often results in an analysis of the whole clause as a simplex, not as a relative clause. Additionally, the functional label of the relative clause carries attachment information so that it is more difficult for the parser to attach them correctly.

For FCR and SGF, the two annotation schemes made different decisions concerning the attachment level of the coordination. In TIGER, the coordination is attached on clause level while TüBa-D/Z coordinates complex fields. As a consequence, the number of possible attachment locations is far higher in TüBa-D/Z and makes it harder for the parser to attach the constituents of the FCR and SGF correctly.

In both models, Unlike Coordinations are exceedingly difficult to parse. The unlexicalized parsers yield slightly better results but the number of CUC sentences is too small to make a strong claim. For PP Attachment, lexicalization plus markovization clearly helps: the Stanford parser outperforms both unlexicalized parsers. The more hierarchical annotation in the TüBa-D/Z annotation scheme makes it easier for the unlexicalized parsers to disambiguate the constituent structure for Noun Attachment while for Verb

	TIGER			TüBa-D/Z			
	Bit	Lop	Stan	Bit	Lop	Stan	Num.
ERC	20	19	19	0	0	3	41
FCR	26	27	23	11	9	13	40
PPN	9	9	16	15	14	14	60
PPV	15	16	18	14	13	18	62
CUC	6	8	5	6	7	5	39
SGF	18	20	20	7	10	8	40

Table 9: Correctly parsed constructions in TIGER and TüBa-D/Z (human evaluation)

Attachment, the flat annotation in TIGER again leaves less space for ambiguity. The results for the different models also reflect the different distribution of Noun versus Verb Attachment in the two treebanks: For approximately 74% of all *noun PP* sequences in TüBa-D/Z, we have Noun Attachment, while in TIGER, only approximately 57% of those PPs are attached to the noun.

In combination with the dependency-based evaluation, the manual evaluation shows that while EVALB and, to a smaller degree, LA favor the TüBa-D/Z annotation scheme, many of the phenomena covered in TePaCoC are easier to parse with TIGER. Obviously, none of the parsers' models are able to cover the hierarchical structure of TüBa-D/Z successfully. The manual evaluation also backs up the dependency-based evaluation and gives more evidence for the already strong suspicion that the PARSEVAL metric, while being a useful tool to assess parser performance for a parser trained on the same training set, is not adequate to give a linguistically motivated assessment of the quality in the parser output.

5. Conclusions

In this paper, we showed how human evaluation of a corpus of complex grammatical constructions allows to detect error types and trace them back to the annotation decision underlying the error. Our main findings are: TIGER benefits from the flat annotation which makes it more transparent and straightforward for the parser to detect constructions like ERC, FCR, or SGF, while TüBa-D/Z suffers from the more hierarchical structure where relevant clues are embedded too deep in the tree for the parser to make use of it. While the additional layer of topological fields in TüBa-D/Z increases the number of possible attachment positions, it also reduces the number of rules in the grammar and improves the learnability especially for small training sets.

Annotated PCFGs such as the ones of Schiehlen (2004) or Versley (2005) successfully modify the treebank so that necessary information is locally available (e.g., by making the topological field nodes transparent to argument structure information). For reasons of simplicity, we did not include annotated PCFGs in our study, but based on these previous results, we expect that the results for annotated PCFGs will be different. We leave this question to future work.

Evidence from the automatic evaluation measures shows that the huge differences in EVALB precision, recall, and F-score between TIGER and TüBa-D/Z are neither reflected in LA results nor in the dependency-based evaluation, the last two being less biased towards a specific treebank annotation scheme. Recent work by Emms (2008) compares the performance of EVALB and variants of EVALB with a tree-distance measure (Zhang and Shasha, 1989), which considers all possible partial mappings between a source and a target tree while preserving left-to-right order and ancestry. Emms applies the alternative measures to the output of 6 off-the-shelf parsers trained on the Penn-II treebank, showing that the ranking of parses from best-to-worst varies for the different measures. He argues that the tree-distance measure is less susceptible to the problem of overrating attachment errors than EVALB. The investigation of the performance of the tree-distance measure on the different German treebanks and its correlation with the dependency-based and human evaluation will provide further insight into the sustainability of the treebank annotations.

Acknowledgments

Ines Rehbein is funded by Science Foundation Ireland GramLab grant 04/IN/I527. Wolfgang Maier and Yannick Versley are funded by the Deutsche Forschungsgemeinschaft (DFG), Wolfgang in the Emmy Noether Program, and Yannick as part of the Collaborative Research Center (SFB) 441.

6. References

Ann Bies and Mohamed Maamouri. 2003. Penn Arabic Treebank guidelines. Technical report, LDC, University of Pennsylvania.

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre, 1995. *Bracketing Guidelines for Treebank II Style Penn Treebank Project*. University of Pennsylvania.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER Treebank. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT 2002)*, pages 24–42, Sozopol, Bulgaria.

Michael Daum, Kilian Foth, and Wolfgang Menzel. 2004. Automatic transformation of phrase treebanks to dependency trees. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal.

Martin Emms. 2008. Tree-distance and some other variants of evalb. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco.

Martin Forst. 2003. Treebank conversion - creating an f-structure bank from the TIGER Corpus. In *Proceedings of the International Lexical Functional Grammar Conference (LFG 2003)*, Saratoga Springs, NY.

Kilian Foth. 2003. Eine umfassende Dependenzgrammatik des Deutschen. Technical report, Fachbereich Informatik, Universität Hamburg.

Chung-Hye Han, Na-Rae Han, and Eon-Suk Ko. 2001. Bracketing guidelines for Penn Korean TreeBank. Technical report, University of Pennsylvania.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 423–430, Sapporo, Japan.

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

Sandra Kübler. 2005. How do treebank annotation schemes influence parsing results? Or how not to compare apples and oranges. In *Proceedings of the 5th International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, pages 293–300, Borovets, Bulgaria.

V. I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics*, 10:707–710.

Dekang Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, pages 1420–1425, Montreal, Canada.

Dekang Lin. 1998. A depedency-based method for evaluating broad-coverage parsers. *Natural Language Engineering*, 4(2):97–114.

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

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 2007)*, pages 630–639, Prague, Czech Republic.

Geoffrey Sampson and Anna Babarczy. 2003. A test of the leaf-ancestor metric for parse accuracy. *Journal of Natural Language Engineering*, 9:365–380.

Michael Schiehlen. 2004. Annotation strategies for probabilistic parsing in German. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2003)*, Geneva, Switzerland.

Anne Schiller, Simone Teufel, and Christine Thielen. 1995. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, Universität Stuttgart, Universität Tübingen.

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

Helmut Schmid. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

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

Yannick Versley. 2005. Parser evaluation across text types. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, Barcelona, Spain.

K. Zhang and D. Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal of Computing*, 18:1245–1262.