Treebank Annotation Schemes and Parser Evaluation for German

Ines Rehbein
NCLT
School of Computing, DCU,
Dublin, Ireland
irehbein@computing.dcu.ie

Josef van Genabith
NCLT,
School of Computing, DCU,
Dublin, Ireland
IBM Dublin Center for Advanced Studies
josef@computing.dcu.ie

Abstract

Recent studies focussed on the question whether less-configurational languages like German are harder to parse than English, or whether the lower parsing scores are an artefact of treebank encoding schemes and data structures, as claimed by Kübler et al. (2006). This claim is based on the assumption that PARSEVAL metrics fully reflect parse quality across treebank encoding schemes. In this paper we present new experiments to test this claim. We use the PARSEVAL metric, the Leaf-Ancestor metric as well as a dependency-based evaluation, and present novel approaches measuring the effect of controlled error insertion on treebank trees and parser output. We also provide extensive past-parsing cross-treebank conversion. The results of the experiments show that, contrary to Kübler et al. (2006), the question whether or not German is harder to parse than English remains undecided.

1 Introduction

A long-standing and unresolved issue in the parsing literature is whether parsing less-configurational languages is harder than e.g. parsing English. German is a case in point. Results from Dubey and Keller (2003) suggest that state-of-the-art parsing scores for German are generally lower than those obtained for English, while recent results from Kübler et al. (2006) raise the possibility that this might be an artefact of particular encoding schemes and data structures of treebanks, which serve as training resources for probabilistic parsers. Kübler (2005) and Maier (2006) show that treebank annotation schemes have considerable influence on parsing results. A comparison of unlexicalised PCFG parsing (Kübler, 2005) trained and evaluated on the German NEGRA (Skut et al., 1997) and the TüBa-D/Z (Telljohann et al., 2004) treebanks using LoPar (Schmid, 2000) shows a difference in parsing results of about 16%, using the PARSEVAL metric (Black et al., 1991). Kübler et al. (2006) conclude that, contrary to what had been assumed, German is not actually harder to parse than English, but that the NEGRA annotation scheme does not support optimal PCFG parsing performance.

Despite being the standard metric for measuring PCFG parser performance, PARSEVAL has been criticised for not representing 'real' parser quality (Carroll et al., 1998; Brisco et al., 2002; Sampson and Babarbczy, 2003). PARSEVAL checks label and wordspan identity in parser output compared to the original treebank trees. It neither weights results, differentiating between linguistically more or less severe errors, nor does it give credit to constituents where the syntactic categories have been recognised correctly but the phrase boundary is slightly wrong.

With this in mind, we question the assumption that the PARSEVAL results for NEGRA and TüBa-D/Z reflect a real difference in quality between the parser output for parsers trained on the two different treebanks. As a consequence we also question the conclusion that PARSEVAL results for German in the same range as the parsing results for the English

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 630–639, Prague, June 2007. ©2007 Association for Computational Linguistics

Penn-II Treebank prove that German is not harder to parse than the more configurational English. To investigate this issue we present experiments on the German TIGER treebank (Dipper et al., 2001) and the TüBa-D/Z treebank. TIGER is based on and extends the NEGRA data and annotation scheme. Our error insertion and past-parsing treebank-encoding experiments experiments show that the differences in parsing results for the two treebanks are not caused by a higher number of errors in the output of the parser trained on the TIGER treebank, but are due to the bias of the PARSEVAL metric towards annotation schemes (such as that of TüBa-D/Z) with a higher ratio of non-terminal/terminal nodes. The experiments also show that compared to PARSEVAL the Leaf-Ancestor metric is somewhat less susceptible to non-terminal/terminal ratios and that contrary to the PARSEVAL results, dependency-based evaluations score TIGER trained parsers higher than TüBa-D/Z trained parsers.

This paper is structured as follows: Section 2 gives an overview of the main features of the two treebanks. Section 3 describes our first experiment, where we systematically insert controlled errors into the original treebank trees and compare the influence of these modifications on the evaluation results in the PARSEVAL metric and the Leaf-Ancestor metric against the original, unmodified trees for both treebanks. In Section 4 we present the second experiment, where we extract an unlexicalised PCFG from each of the treebanks. Then we convert the output of the PCFG parser trained on the TüBa-D/Z into a TIGER-style format and evaluate the converted trees. In Section 5 we present a dependency-based evaluation and compare the results to the results of the two other measures. The last section concludes.

2 The TIGER Treebank and the TüBa-D/Z

The two German treebanks used in our experiments are the TIGER Treebank (Release 2) and the Tüba-D/Z (Release 2). The TüBa-D/Z consists of approximately 22 000 sentences, while the TIGER Treebank is much larger with more than 50 000 sentences. Both treebanks contain German newspaper text and are annotated with phrase structure and dependency (functional) information. Both treebanks use the Stuttgart Tübingen POS Tag Set (Schiller et al., 95). TIGER uses 49 different grammatical function labels, while the TüBa-D/Z utilises only 36 function labels. For the encoding of phrasal node categories the TüBa-D/Z uses 30 different categories, the TIGER Treebank uses a set of 27 category labels.

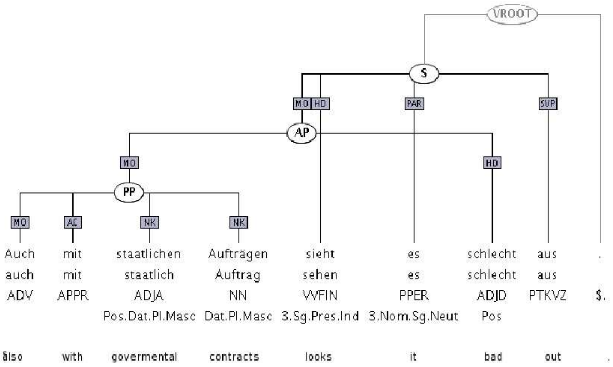
Other major differences between the two treebanks are: in the Tiger Treebank long distance dependencies are expressed through crossing branches (Figure 1), while in the TüBa-D/Z the same phenomenon is expressed with the help of grammatical function labels (Figure 2), where the node label V-MOD encodes the information that the PP modifies the verb. The annotation in the Tiger Treebank is rather flat and allows no unary branching, whereas the nodes in the TüBa-D/Z do contain unary branches and a more hierarchical structure, resulting in a much deeper tree structure than the trees in the Tiger Treebank. This results in an average higher number of nodes per sentence for the TüBa-D/Z. Table 1 shows the differences in the ratio of nodes for the Tiger treebank and the TüBa-D/Z.

	phrasal nodes/sent	phrasal nodes/word	words /sent
TIGER	8.29	0.47	17.60
TüBa-D/Z	20.69	1.20	17.27

Table 1: Average number of phrasal nodes/words in TIGER and TüBa-D/Z

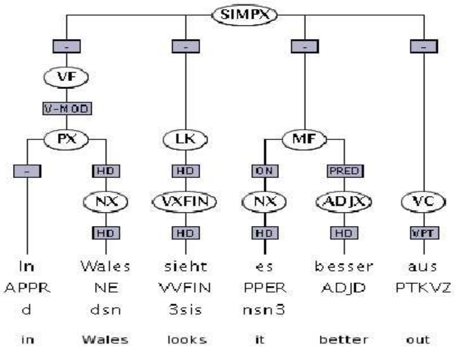
Figures 1 and 2 also illustrate the different annotation of PPs in both annotation schemes. In the Tiger treebank the internal structure of the PP is flat and the adjective and noun inside the PP are directly attached to the PP, while the TüBa-D/Z is more hierarchical and inserts an additional NP node.

Another major difference is the annotation of topological fields in the style of Drach (1937) and Höhle (1986) in the TüBa-D/Z. The model captures German word order, which accepts three possible sentence configurations (verb first, verb second and verb last), by providing fields like the initial field (VF), the middle field (MF) and the final field (NF). The fields are positioned relative to the verb, which can fill in the left (LK) or the right sentence bracket (VC). The ordering of topological fields is determined by syntactic constraints.


VROOT
S
MO
HD
PAR
SVP
AP
MO
HD
PP
MO
AC
NK
NK
Auch auch ADV also
mit mit APPR with
staatlichen staatlich ADJA Pos.Dat.Pl.Masc govermental
Aufträgen Auftrag NN Dat.Pl.Masc contracts
sieht sehen VVFIN 3.Sg.Pres.Ind looks
es es PPER 3.Nom.Sg.Neut it
schlecht schlecht ADJD Pos bad
aus aus PTKVZ out
. $. .

Auch mit staatlichen Aufträgen sieht es schlecht aus.
"It also looks bad for public contracts."

Figure 1: TIGER treebank tree



In Wales sieht es besser aus.
"Things seem better in Wales."

Figure 2: TüBa-D/Z treebank tree

2.1 Differences between TIGER and NEGRA

To date, most PCFG parsing for German has been done using the NEGRA corpus as a training resource. The flat annotation scheme of the TIGER treebank is based on the NEGRA annotation scheme, but it also employs some important extensions, which include the annotation of verb-subcategorisation, appositions and parentheses, coordinations and the encoding of proper nouns (Brants et al., 2002).

3 Treebank Preprocessing: Converting TIGER Graphs into CFG Trees

The sentences in the TIGER treebank are represented as graphs with LDDs expressed through crossing branches. Before being able to insert errors or extract a PCFG we had to resolve these crossing branches in the TIGER treebank. This was done by attaching the non-head child nodes higher up in the tree, following Kübler (2006). For the graph in Figure 1 this would mean that the modifying PP *"Auch mit staatlichen Aufträgen"* (also for public contracts) was attached directly to the S node, while the head of the adjectival phrase (AP) remained in it's original position. As a side effect this leads to the creation of some unary nodes in the TIGER trees. We also inserted a virtual root node and removed all functional labels from the TIGER and TüBa-D/Z trees.

4 Experiment I

Experiment I is designed to assess the impact of identical errors on the two treebank encoding schemes and the PARSEVAL[1] and Leaf-Ancestor evaluation metrics.

4.1 Experimental Setup

The TIGER treebank and the TüBa-D/Z both contain newspaper text, but from different German newspapers. To support a meaningful comparison we have to compare similar sentences from both treebanks. In order to control for similarity we selected all sentences of length $10 \leq n \leq 40$ from both treebanks. For all sentences with equal length we computed the average number of prepositions, determiners, nouns (and related POS such as proper names and personal pronouns), interrogative pronouns, finite verbs, infinite verbs, past participles and imperative verb forms. For each sentence length we selected all sentences from both treebanks which showed an average for each of the POS listed above which did not deviate more than 0.8 from the average for all sentences for this particular sentence length. From this set we randomly selected 1024 sentences for each of the treebanks. This results in two test sets, comparable in word length, syntactic structure and complexity. Table 2 shows the ratio of phrasal versus terminal nodes in the test sets.

We then inserted different types of controlled errors automatically into the *original* treebank trees in our test sets and evaluated the modified trees against

[1] In all our experiments we use the `evalb` metric (Sekine and Collins, 1997), the most commonly used implementation of the PARSEVAL metric.

	phrasal nodes/sent	phrasal nodes nodes/word	words /sent
TIGER	6.97	0.48	14.49
TüBa-D/Z	19.18	1.30	14.75

Table 2: Average number of phrasal nodes/words in the TIGER and TüBa-D/Z test set

the original treebank trees, in order to assess the impact of similar (controlled for type and number) errors on the two encoding schemes.

4.2 Error Insertion

The errors fall into three types: attachment, span and labeling (Table 3). We carried out the same number of error insertions in both test sets.

	Error description
ATTACH I	Attach PPs inside an NP one level higher up in the tree
ATTACH II	Change verb attachment to noun attachment for PPs on sentence level, inside a VP or in the MF (middle field)
LABEL I	Change labels of PPs to NP
LABEL II	Change labels of VPs to PP
SPAN I	Include adverb to the left of a PP into the PP
SPAN II	Include NN to the left of a PP into the PP
SPAN III	Combination of SPANI and SPANII

Table 3: Description of inserted error types

4.3 Results for Error Insertion for the Original Treebank Trees

Table 4 shows the impact of the error insertion into the original treebank trees on PARSEVAL results, evaluated against the gold trees. PARSEVAL results in all experiments report labelled precision and recall. The first error (PP attachment I, 85 insertions in each test set) leads to a decrease in f-score of 1.16 for the TIGER test set, while for the TüBa-D/Z test set the same error only caused a decrease of 0.43. The effect remains the same for all error types and is most pronounced for the category label errors, because the frequency of the labels resulted in a large number of substitutions. The last row lists the total weighted average for all error types, weighted with respect to their frequency of occurrence in the test sets.

Table 4 clearly shows that the PARSEVAL measure punishes the TIGER treebank annotation scheme to a greater extent, while the same number and type of errors in the TüBa-D/Z annotation scheme does not have an equally strong effect on PARSEVAL results for similar sentences.

	TIGER	TüBa	# errors
PP attachment I	98.84	99.57	85
PP attachment II	98.75	99.55	89
Label I	80.02	92.73	1427
Label II	93.00	97.45	500
SPAN I	99.01	99.64	71
SPAN II	97.47	99.08	181
SPAN III	96.51	98.73	252
total weighted ave.	87.09	95.30	

Table 4: f-score for PARSEVAL results for error insertion in the original treebank trees

4.4 Discussion: PARSEVAL and LA

Experiment I shows that the gap between the PARSEVAL results for the two annotation schemes does not reflect a difference in quality between the trees. Both test sets contain the same number of sentences with the same sentence length and are equivalent in complexity and structure. They contain the same number and type of errors. This suggests that the difference between the results for the TIGER and the TüBa-D/Z test set are due to the higher ratio of non-terminal/terminal nodes in the TüBa-D/Z trees (Table 1).

In order to obtain an alternative view on the quality of our annotation schemes we used the leaf-ancestor (LA) metric (Sampson and Babarbczy, 2003), a parser evaluation metric which measures the similarity of the path from each terminal node in the parse tree to the root node. The path consists of the sequence of node labels between the terminal node and the root node, and the similarity of two paths is calculated by using the Levenshtein distance (Levenshtein, 1966). Table 5 shows the results for the leaf-ancestor evaluation metric for our error insertion test sets. Here the weighted average results for the two test sets are much closer to each other (94.98 vs. 97.18 as against 87.09 vs. 95.30). Only the label errors, due to the large numbers, show a significant difference between the two annotation schemes. Tables 4 and 5 show that compared to PARSEVAL the LA metric is somewhat less sensitive to the nonterminal/terminal ratio.

Figure 3 illustrates the different behaviour of the

	TIGER	**TüBa**	**# errors**
PP attachment I	99.62	99.70	85
PP attachment II	99.66	99.78	89
Label I	92.45	95.24	1427
Label II	96.05	99.28	500
SPAN I	99.82	99.84	71
SPAN II	99.51	99.77	181
SPAN III	99.34	99.62	252
total weighted ave.	94.98	97.18	

Table 5: LA results for error insertion in the original treebank trees

two evaluation metrics with respect to an example sentence.

Sentence 9:
Die Stadtverwaltung von Venedig hat erstmals streunende Katzen gezählt.
"For the first time the city council of Venice has counted straying cats."

```
(TOP
    (S
      (NP
          (ART Die [the] )
          (NN Stadtverwaltung [city counsil] )
          (PP
              (APPR von [of] )
              (NE Venedig [Venice] )
          )
      )
      (VAFIN hat [has] )
      (VP
          (ADV erstmals [for the first time] )
          (NP
              (ADJA streunende [straying] )
              (NN Katzen [cats] )
          )
          (VVPP gezählt [counted] )
      )
    )
    ($. .)
)
```

Figure 3: Sentence 9 from the TIGER Test Set

Table 6 shows that all error types inserted into Sentence 9 in our test set result in the same evaluation score for the PARSEVAL metric, while the LA metric provides a more discriminative treatment of PP attachment errors, label errors and span errors for the same sentence (Table 6). However, the differences in the LA results are only indirectly caused by the different error types. They actually reflect the number of terminal nodes affected by the error insertion. For Label I and II the LA results vary considerably, because the substitution of the PP for an NP (Label I) in Figure 3 affects two terminal nodes only (PP *von* [of] *Venedig* [Venice]), while the change of the VP into a PP (Label II) alters the paths of four terminal nodes (VP *erstmals* [for the first time] *streunende* [straying] *Katzen* [cats] *gezählt* [counted]) and therefore has a much greater impact on the overall result for the sentence.

ERROR	PARSEVAL	LA
PP attachment I	83.33	96.30
Label I	83.33	96.00
Label II	83.33	91.00
SPAN II	83.33	96.40

Table 6: Evaluation results for Sentence 9

The TüBa-D/Z benefits from its overall higher ratio of nodes per sentence, resulting in a higher ratio of non-terminal/terminal nodes per phrase and the effect, that the inserted label error affects a smaller number of terminal nodes than in the TIGER test set for LA testing.

5 Experiment II

Kübler (2005) and Maier (2006) assess the impact of the different treebank annotation schemes on PCFG parsing by conducting a number of modifications converting the TüBa-D/Z into a format more similar to the NEGRA (and hence TIGER) treebank. After each modification they extract a PCFG from the modified treebank and measure the effect of the changes on parsing results. They show that with each modification transforming the TüBa-D/Z into a more NEGRA-like format the parsing results also become more similar to the results of the NEGRA treebank, i.e. the results get worse. Maier takes this as evidence that the TüBa-D/Z is more adequate for PCFG parsing. This assumption is based on the belief that PARSEVAL results fully reflect parse quality across different treebank encoding schemes. This is not always true, as shown in Experiment I.

In our second experiment we crucially change the order of events in the Kübler (2005), Maier (2006) and Kübler et al. (2006) experiments: We first extract an unlexicalised PCFG from each of the original treebanks. We then transform the output of the parser trained on the TüBa-D/Z into a format more similar to the TIGER Treebank. In contrast to Kübler (2005) and Maier (2006), who converted the treebank before extracting the grammars in order to measure the impact of single features like topological fields or unary nodes on PCFG parsing, we convert the trees in the parser *output* of a parser trained on the *original* unconverted treebank resources. *This allows us to preserve the basic syntactic structure and also the errors present in the output trees resulting from a potential bias in the original treebank training resources*. The results for the original parser output evaluated against the unmodified gold trees should not be crucially different from the results for the modified parser output evaluated against the modified gold trees.

5.1 Experimental Setup

For Experiment II we trained BitPar (Schmid, 2004), a parser for highly ambiguous PCFG grammars, on the two treebanks. The TüBa-D/Z training data consists of the 21067 treebank trees not included in the TüBa-D/Z test set. Because of the different size of the two treebanks we selected 21067 sentences from the TIGER treebank, starting from sentence 10000 (and excluding the sentences in the TIGER test set).

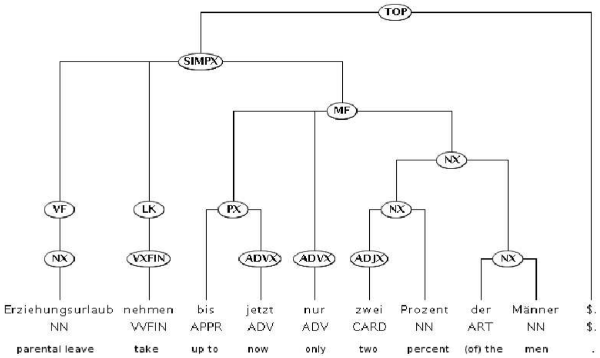
Before extracting the grammars we resolved the crossing branches in the TIGER treebank as described in Section 3. After this preprocessing step we extracted an unlexicalised PCFG from each of our training sets. Our TIGER grammar has a total of 21163 rule types, while the grammar extracted from the TüBa-D/Z treebank consists of 5021 rules only. We parsed the TIGER and TüBa-D/Z test set with the extracted grammars, using the gold POS tags for parser input. We then automatically converted the TüBa-D/Z output to a TIGER-like format and compare the evaluation results for the unmodified trees against the gold trees with the results for the converted parser output against the converted gold trees.

5.2 Converting the TüBa-D/Z Trees

The automatic conversion of the TüBa-D/Z-style trees includes the removal of topological fields and unary nodes as well as the deletion of NPs inside of PPs, because the NP child nodes are directly attached to the PP in the TIGER annotation scheme. As a last step in the conversion process we adapted the TüBa-D/Z node labels to the TIGER categories.

5.2.1 The Conversion Process: An Example

We demonstrate the conversion process using an example sentence from the TüBa-D/Z test set (Figure 4). The converted tree is given in Figure 5: topological fields, here VF (initial field), MF (middle field) and LK (left sentence bracket), as well as unary nodes have been removed. The category labels have been changed to TIGER-style annotation.



Erziehungsurlaub nehmen bisher nur zwei Prozent der Männer.
"Until now only two percent of the men take parental leave."

Figure 4: Original TüBa-D/Z-style gold tree

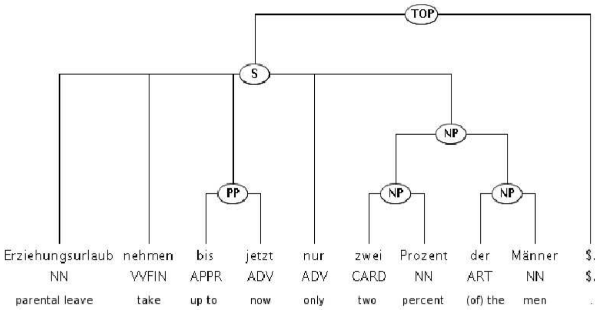


Figure 5: Converted TIGER-style gold tree

Figure 6 shows the unmodified parser output from the TüBa-D/Z trained grammar for the same string. The parser incorrectly included all adverbs inside an NP governed by the PP, while in the gold tree (Figure 4) both adverbs are attached to the PP. The modified parser output is shown in Figure 7.

5.3 Results for Converted Parser Output

We applied the conversion method described above to the original trees and the parser output for the sentences in the TIGER and the TüBa-D/Z test sets. Table 7 shows PARSEVAL and LA results for the modified trees, evaluating the converted parser output

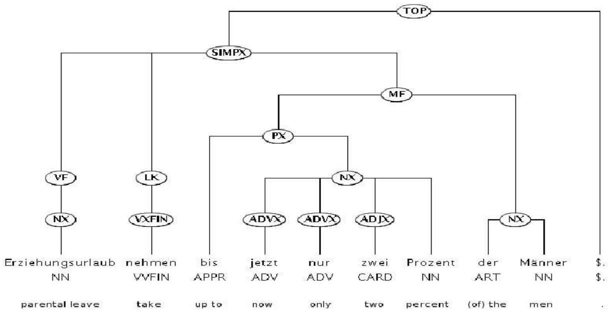

TOP
SIMPX
MF
PX
VF
LK
NX
NX
VXFIN
ADVX
ADVX
ADJX
NX
Erziehungsurlaub nehmen bis jetzt nur zwei Prozent der Männer $.
NN VVFIN APPR ADV ADV CARD NN ART NN $.
parental leave take up to now only two percent (of) the men .

Figure 6: Parser output (TüBa-D/Z grammar)

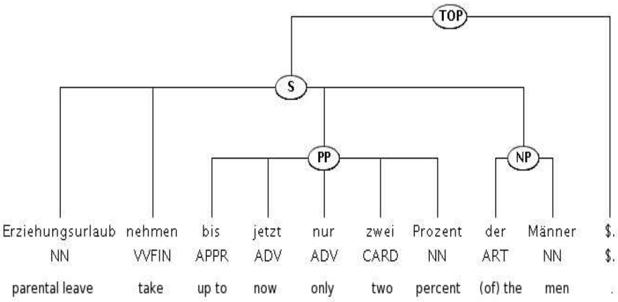

TOP
S
PP
NP
Erziehungsurlaub nehmen bis jetzt nur zwei Prozent der Männer $.
NN VVFIN APPR ADV ADV CARD NN ART NN $.
parental leave take up to now only two percent (of) the men .

Figure 7: Converted parser output (TüBa-D/Z)

	EVALB			LA
	prec.	**recall**	**f-sco.**	**avg.**
TIGER	83.54	83.65	83.59	94.69
no Unary	84.33	84.48	84.41	94.83
TüBa-D/Z	92.59	89.79	91.17	94.23
TüBa-D/Z → TIGER				
no Top	92.38	88.76	90.53	93.93
no Unary	89.96	85.67	87.76	93.59
no Top + no U.	88.44	82.24	85.23	92.91
no Top + no U. + no NP in PP	87.15	79.52	83.16	92.47

Table 7: The impact of the conversion process on PARSEVAL and LA

for each treebank against the converted gold trees of the same treebank. Due to the resolved crossing branches in the TIGER treebank we also have some unary nodes in the TIGER test set. Their removal surprisingly improves both PARSEVAL and LA results. For the TüBa-D/Z all conversions lead to a decrease in precision and recall for the PARSEVAL metric. Converting the trees parsed by the TüBa-D/Z grammar to a TIGER-like format produces an f-score which is slightly lower than that for the TIGER trees. The same is true for the LA metric, but not to the same extent as for PARSEVAL. The LA metric also gives slightly better results for the original TIGER trees compared to the result for the unmodified TüBa-D/Z trees.

The constant decrease in PARSEVAL results for the modified trees is consistent with the results in Kübler et al. (2005), but our conclusions are slightly different. Our experiment shows that the TüBa-D/Z annotation scheme does not generally produce higher quality parser output, but that the PARSEVAL results are highly sensitive to the ratio of non-terminal/terminal nodes. However, the parser output for the grammar trained on the TüBa-D/Z yields a higher precision in the PARSEVAL metric against the TüBa-D/Z gold trees than the parser output of the TIGER grammar against the TIGER gold trees. For PARSEVAL recall, the TIGER grammar gives better results.

6 Experiment III

In Experiment I and II we showed that the tree-based PARSEVAL metric is not a reliable measure for comparing the impact of different treebank annotation schemes on the quality of parser output and that the issue, whether German is harder to parse than English, remains undecided. In Experiment III we report a dependency-based evaluation and compare the results to the results of the other metrics.

6.1 Dependency-Based (DB) Evaluation

The dependency-based evaluation used in the experiments follows the method of Lin (1998) and Kübler and Telljohann (2002), converting the original treebank trees and the parser output into dependency relations of the form WORD POS HEAD. Functional labels have been omitted for parsing, therefore the dependencies do not comprise functional information. Figure 8 shows the original TIGER Treebank representation for the CFG tree in Figure 3. Square boxes denote grammatical functions. Figure 9 shows the dependency relations for the same tree, indicated by labelled arrows. Converted into a WORD POS HEAD triple format the dependency tree looks as follows (Table 8).

Following Lin (1998), our DB evaluation algorithm computes precision and recall:

- **Precision**: the percentage of dependency relationships in the parser output that are also

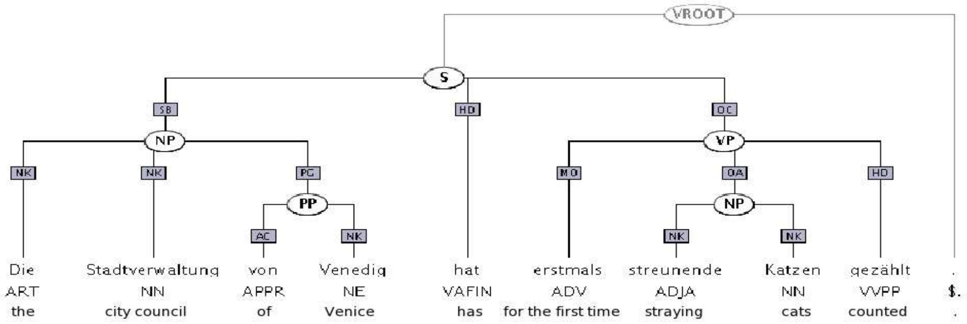
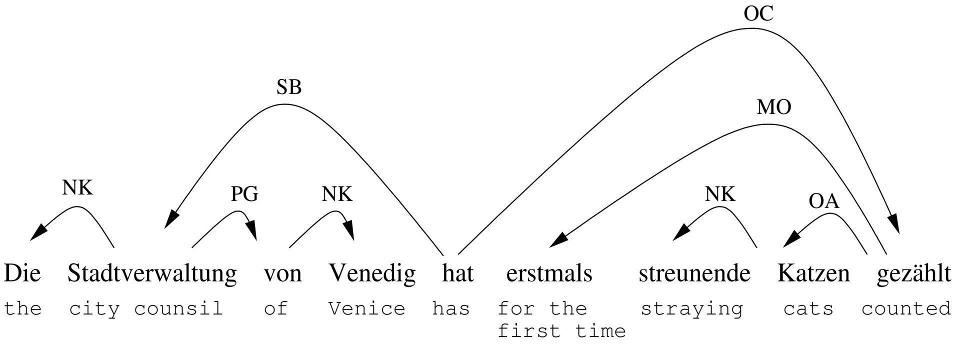


Figure 8: TIGER treebank representation for Figure 3



"For the first time the city counsil of Venice has counted straying cats."

Figure 9: Dependency relations for Figure 8

found in the gold triples

- **Recall**: the percentage of dependency relationships in the gold triples that are also found in the parser output triples.

WORD		**POS**	**HEAD**
Die	[the]	ART	Stadtverwaltung
Stadtverwaltung	[city counsil]	NN	hat
von	[of]	APPR	Stadtverwaltung
Venedig	[Venice]	NE	von
hat	[has]	VAFIN	-
erstmals	[for the first time]	ADV	gezählt
streunende	[straying]	ADJA	Katzen
Katzen	[cats]	NN	gezählt
gezählt	[counted]	VVPP	hat

Table 8: Dependency triples for Figure 9

We assessed the quality of the automatic conversion methodology by converting the 1024 original trees from each of our test sets into dependency relations, using the functional labels in the original trees to determine the dependencies. Topological fields in the TüBa-D/Z test set have been removed before extracting the dependency relationships.

We then removed all functional information from the trees and converted the stripped trees into dependencies, using heuristics to find the head. We evaluated the dependencies for the stripped gold trees against the dependencies for the original gold trees including functional labels and obtained an f-score of 99.64% for TIGER and 99.13% for the TüBa-D/Z dependencies. This shows that the conversion is reliable and not unduly biased to either the TIGER or TüBa-D/Z annotation schemes.

6.2 Experimental Setup

For Experiment III we used the same PCFG grammars and test sets as in Experiment II. Before extracting the dependency relationships we removed the topological fields in the TüBa-D/Z parser output. As shown in Section 6.1, this does not penalise the dependency-based evaluation results for the TüBa-D/Z. In contrast to Experiment II we used raw text as parser input instead of the gold POS tags, allowing a comparison with the gold tag results in Table 7.

6.3 Results

Table 9 shows the evaluation results for the three different evaluation metrics. For the DB evaluation the parser trained on the TIGER training set achieves about 7% higher results for precision and recall than the parser trained on the TüBa-D/Z. This result is clearly in contrast to the PARSEVAL scores, which show higher results for precision and recall for the TüBa-D/Z. But contrary to the PARSEVAL results on gold POS tags as parser input (Table 7), the gap between the results for TIGER and TüBa-D/Z is not as wide as before. PARSEVAL gives a labelled bracketing f-score of 81.12% (TIGER) and 85.47% (TüBa-D/Z) on raw text as parser input, while the results on gold POS tags are more distinctive with an f-score of 83.59% for TIGER and 91.17% for TüBa-D/Z. The LA results again give better scores to the TIGER parser output, this time the difference is more pronounced than for Experiment II (Table 7).

	Dependencies		PARSEVAL		LA
	Prec	Rec	Prec	Rec	Avg
TIGER	85.71	85.72	81.21	81.04	93.88
TüBa	76.64	76.63	87.24	83.77	92.58

Table 9: Parsing results for three evaluation metrics

The considerable difference between the results for the metrics raises the question which of the metrics is the most adequate for judging parser output quality across treebank encoding schemes.

7 Conclusions

In this paper we presented novel experiments assessing the validity of parsing results measured along different dimensions: the tree-based PARSEVAL metric, the string-based Leaf-Ancestor metric and a dependency-based evaluation. By inserting controlled errors into gold treebank trees and measuring the effects on parser evaluation results we gave new evidence for the downsides of PARSEVAL which, despite severe criticism, is still the standard measure for parser evaluation. We showed that PARSEVAL cannot be used to compare the output of PCFG parsers trained on different treebank annotation schemes, because the results correlate with the ratio of non-terminal/terminal nodes. Comparing two different annotation schemes, PARSEVAL consistently favours the one with the higher node ratio.

We examined the influence of treebank annotation schemes on unlexicalised PCFG parsing, and rejected the claim that the German TüBa-D/Z treebank is more appropriate for PCFG parsing than the German TIGER treebank and showed that converting the TüBa-D/Z trained parser output to a TIGER-like format leads to PARSEVAL results slightly worse than the ones for the TIGER treebank trained parser. Additional evidence comes from a dependency-based evaluation, showing that, for the output of the parser trained on the TIGER treebank, the mapping from the CFG trees to dependency relations yields better results than for the grammar trained on the TüBa-D/Z annotation scheme, even though PARSEVAL scores suggest that the TIGER-based parser output trees are substantial worse than TüBa-D/Z-based parser output trees.

We have shown that different treebank annotation schemes have a strong impact on parsing results for similar input data with similar (simulated) parser errors. Therefore the question whether a particular language is harder to parse than another language or not, can not be answered by comparing parsing results for parsers trained on treebanks with different annotation schemes. Comparing PARSEVAL-based parsing results for a parser trained on the TüBa-D/Z or TIGER to results achieved by a parser trained on the English Penn-II treebank (Marcus et al., 1994) does not provide conclusive evidence about the parsability of a particular language, because the results show a bias introduced by the combined effect of annotation scheme and evaluation metric. This means that the question whether German is harder to parse than English, is still undecided. A possible way forward is perhaps a dependency-based evaluation of TIGER/TüBa-D/Z with Penn-II trained grammars for 'similar' test and training sets and cross-treebank and -language controlled error insertion experiments. Even this is not entirely straightforward as it is not completely clear what constitutes 'similar' test/training sets across languages. We will attempt to pursue this in further research.

Acknowledgements

We would like to thank the anomymous reviewers for many helpful comments. This research has been supported by a Science Foundation Ireland grant 04|IN|I527.

References

Black, E., S. P. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. P. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. *In Proceedings DARPA Speech and Natural Language Workshop, Pacific Grove, CA*, pp. 306-311.

Brants, Sabine, and Silvia Hansen. 2002. Developments in the TIGER Annotation Scheme and their Realization in the Corpus. *In Proceedings of the Third Conference on Language Resources and Evaluation (LREC 2002)*, pp. 1643-1649 Las Palmas.

Briscoe, E. J., J. A. Carroll, and A. Copestake. 2002. Relational evaluation schemes. *In Proceedings Workshop 'Beyond Parseval - towards improved evaluation measures for parsing systems', 3rd International Conference on Language Resources and Evaluation*, pp. 4-38. Las Palmas, Canary Islands.

Carroll, J., E. Briscoe and A. Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. *In Proceedings of the 1st International Conference on Language Resources and Evaluation*, Granada, Spain. 447-454.

Dipper, S., T. Brants, W. Lezius, O. Plaehn, and G. Smith. 2001. The TIGER Treebank. *In Third Workshop on Linguistically Interpreted Corpora LINC-2001*, Leuven, Belgium.

Drach, Erich. 1937. *Grundgedanken der Deutschen Satzlehre.* Frankfurt/M.

Dubey, A., and F. Keller. 2003. Probabilistic parsing for German using sisterhead dependencies. *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan.

Höhle, Tilman. 1998. Der Begriff "Mittelfeld", Anmerkungen über die Theorie der topologischen Felder. *In Akten des Siebten Internationalen Germansitenkongresses 1985*, pages 329-340, Göttingen, Germany.

Kübler, Sandra, and Heike Telljohann. 2002. Towards a Dependency-Oriented Evaluation for Partial Parsing. *In Proceedings of Beyond PARSEVAL – Towards Improved Evaluation Measures for Parsing Systems (LREC 2002 Workshop)*, Las Palmas, Gran Canaria, June 2002.

Lin, Dekang. 1998. A dependency-based method for evaluating broad-coverage parsers. *Natural Language Engineering*, 1998.

Kübler, Sandra. 2005. How Do Treebank Annotation Schemes Influence Parsing Results? Or How Not to Compare Apples And Oranges. *In Proceedings of FANLP 2005)*, Borovets, Bulgaria, September 2005.

Kübler, Sandra, Erhard 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, EMNLP 2006)*, Sydney, Australia, July 2006.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics - Doklady*, 10.707-10 (translation of Russian original published in 1965).

Maier, Wolfgang. 2006. Annotation Schemes and their Influence on Parsing Results. *In Proceedings of the COLING/ACL 2006 Student Research Workshop)*, Sydney, Australia, July 2006.

Marcus, M., G. Kim, M. A. Marcinkiewicz, R. MacIntyre, M. Ferguson, K. Katz and B. Schasberger. 1994. The Penn Treebank: Annotating Predicate Argument Structure. *In Proceedings of the ARPA Human Language Technology Workshop*, Princeton, NJ.

Sampson, Geoffrey, and Anna Babarczy. 2003. A test of the leaf-ancestor metric for parse accuracy. *Natural Language Engineering*, 9 (4):365-380.

Schmid, Helmut. 2000. LoPar: Design and Implementation. *Arbeitspapiere des Sonderforschungsbereiches 340, No. 149*, IMS Stuttgart, July 2000.

Schmid, Helmut. 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.

Sekine, S. and M. J. Collins. 1997. The evalb software. http://nlp.cs.nyu.edu/evalb/

Skut, Wojciech, Brigitte Krann, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. *In Proceedings of ANLP 1997*, Washington, D.C.

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

Schiller, Anne, Simone Teufel, and Christine Thielen. 1995. *Guidelines fr das Tagging deutscher Textcorpora mit STTS.* Technical Report, IMS-CL, University Stuttgart, 1995.