Parser evaluation across Text Types

Yannick Versley
Seminar für Sprachwissenschaft
Universität Tübingen
E-mail: versley@sfs.uni-tuebingen.de

1 Introduction

When a statistical parser is trained on one treebank, one usually tests it on another portion of the same treebank, partly due to the fact that a comparable annotation format is needed for testing. But the user of a parser may not be interested in parsing sentences from the same newspaper all over, or even wants syntactic annotations for a slightly different text type. Gildea (2001) for instance found that a parser trained on the WSJ portion of the Penn Treebank performs less well on the Brown corpus (the subset that is available in the PTB bracketing format) than a parser that has been trained only on the Brown corpus, although the latter one has only half as many sentences as the former. Additionally, a parser trained on both the WSJ and Brown corpora performs less well on the Brown corpus than on the WSJ one.[1]

This leads us to the following questions that we would like to address in this paper:

- Is there a difference in usefulness of techniques that are used to improve parser performance between the same-corpus and the different-corpus case?
- Are different types of parsers (rule-based and statistical) equally sensitive to corpus variation?

[1] Ratnaparkhhi (1998) made a similar experiment on the "Magazine & Journal Articles", "General Fiction" and "Adventure Fiction" sections from the Brown Corpus and notes that the 5-7% loss in accuracy for his maximum-entropy parser cannot be accounted for by inherent parsing difficulty since training on the same amount of text yields similar (within 1%) results for the same-corpus case. Roark and Bacchiani find in their adaptation experiments that their parser on the Brown corpus and testing on WSJ texts yields a 9% loss in comparison to testing on the Brown corpus and a 10% loss in comparison to training on a similar amount of WSJ text.

To achieve this, we compared the quality of the parses of a hand-crafted constraint-based parser and a statistical PCFG-based parser that was trained on a treebank of German newspaper text.

2 The parsing models

Our reference regarding both target annotations and parse quality is the work of Foth et al. (2004), who evaluated their Weighted Constraint Dependency Grammar (WCDG) parser for German against a range of different text types including trivial literature, law text, appointment dialogues, online news and serious literature. They found that the performance of their parser varied considerably between the genres (from 78% unlabeled recall for serious literature to 93% for trivial literature).

The WCDG parser is a combination of an optimizer for weighted constraints over (not necessarily projective) dependency trees using a local search algorithm (a variant of taboo search) and a grammar consisting of handwritten weighted constraints in a declarative language, resulting in a mechanism that is both more flexible and computationally intensive than the lexicalized PCFG approach that until quite recently constituted the state of the art in English statistical parsing.[2]

2.1 The statistical parser

In the spirit of Klein and Manning (2003), as well as Schiehlen (2004), we use an unlexicalized PCFG model that is obtained by enriching the original trees of the treebank (the TüBa-D/Z treebank of written German; see Telljohann et al., 2003) with additional information that would otherwise get lost due to the independence assumption inherent in PCFGs and use the annotated trees to train the model. After parsing a sentence using BitPar, a fast CKY parser (Schmid, 2004), we use the sequence of annotated node labels to (re-)construct edge labels and subsequently discard the additional annotations, resulting in a tree that is (mostly) the same as it would have been in the original treebank.

2.2 Converting to dependencies

In order to be able to compare the parse results with those of Foth et al. (2004), we use a set of handcrafted rules that transform the original TüBa-D/Z annotations to dependencies in the format Foth et al.'s WCDG parser, similar to the tool used to

[2]Note that recent work on reranking, the latest being Charniak and Johnson (2005), emphasizes the use of features that cannot be handled efficiently by a lexicalized PCFG-like grammar.

convert the nonprojective version of the Negra annotations to dependencies (Daum et al., 2004), but geared towards the format of TüBa-D/Z.

The main advantage of this approach is a greater independence from the original phrase structure in the corpus, since the number of dependency relations does not vary as much between different annotation schemes as the number and kind of constituents in a phrase-structure-based annotation scheme (consider, for example, the introduction of NP constituents inside PPs or unary nodes for certain preterminals in a treebank that doesn't normally contain them, or the introduction of topological fields). Besides this, syntactic dependency relations are much closer to the semantic predicate-argument structure that is needed for the semantic interpretation of a sentence or any approximation thereof.

Some features of TüBa-D/Z make the task of extracting dependencies more difficult than with the structure of the Negra treebank; the nodes inside of the Vorfeld and Mittelfeld topological fields depend on either the main verb or the finite verb, which may be the same word, different words inside the same node or even inside different nodes, necessitating a slightly more complex approach than in Daum's DepSy tool.

Nonprojective structure beyond the topological fields in TüBa-D/Z is annotated partly in the edge labels and partly in secondary edges. For example the nonprojectively attached relative clause

(1) Ich habe [$_{\text{NX.OA}}$ den Staubsauger] gekauft, [$_{\text{R-SIMPX.OA-MOD}}$ den Du Dir gewünscht hast].

is marked as a modifier of the accusative object, whereas in the phrase

(2) Ich habe [$_{\text{PX.V-MOD}}$ von [$_{\text{NX}}$ dem Kuchen]] gegessen, [$_{\text{R-SIMPX.MOD}}$ der gestern übriggeblieben ist].

a secondary edge would link the NP "dem Kuchen" to its relative clause adjunct.

The dependency converter is able to make use of this information to reattach relative clauses and other shifted material to its governor in the case of the original treebank annotations. In the case of parsed sentences, however, no secondary edges are available and the decoration from the PCFG model contains no useful information about the attachment. It would be possible to reattach the shifted material via a postprocessing step, but for the sake of simplicity (and since relative clauses constitute less than 2% of the dependencies), this was not done here.

A comparison of the automatically converted dependencies against manually annotated ones uncovered several conceptual differences between the TüBa-D/Z annotation and the WCDG annotation:

- Even for commonly used verbs, the question of what is and what is not a PP argument (as opposed to an adjunct) is handled differently by TüBa-D/Z and the WCDG annotation. Thus, we conflated the "PP" and "OBJP" labels in the dependency model when evaluating the parser.

- The distinction between the subject and the predicate in a predicative construction (as in “Peter is a fool”) is different the TüBa-D/Z and the WCDG annotation guidelines (Telljohann et al., 2003; Foth, 2003). The former recommends that an NP in the Vorfeld which has the right case and agrees with the verb is always to be labeled as the subject, while the WCDG annotation guide recommends to consider variations of the sentence in question to carry out the decision on semantic grounds. For comparison purposes, we conflated the “SUBJ” and “PRED” labels in the dependency evaluation.
- Adverbs and prepositional phrases that are attached to a noun phrase in the WCDG version are quite often attached to the dominating verb in TüBa-D/Z. The latter is based on a “high attachment principle” stating that “in the case of syntactic and semantic ambiguity in the attachment of modifiers such ambiguous modifiers are attached to the highest possible level in a tree structure”.
- There are three roughly different possibilities for nominal postmodification of noun phrases:

  (3) a. [$_{NX}$ Harald], [$_{NX}$ der Wurm]
  b. [$_{NX}$ IBM] [$_{NX}$ Deutschland]
  c. [$_{NX}$ der Kurier] [$_{NX}$ des Zaren]

  (a), which could be called ‘true’ apposition, involves two complementing descriptions for a single entity, (b) introduces a noun postmodifier that is not a complete description but rather an adjoined part of a name, and (c) is a possessive construction involving a genitive-marked postmodifier that indicates the possessor.

  The WCDG annotation uses the APP label for both (a) and (b), and the GMOD (genitive modifier) label for possessive postmodification (c), while TüBa-D/Z distinguish between true apposition (a) on one side and possessive and other postmodification (b and c) on the other side. This mismatch means that distinguishing the APP and GMOD labels when converting to dependencies crucially depends on case information being available, and is nearly impossible in the case of a genitive NP being modified.

2.3 Modification of the treebank grammar

Unlike the Negra treebank, the TüBa-D/Z treebank has detailed morphological (number/gender/case/person) information for all sentences. Moreover, it does not use different categories for coordinated phrases, includes adjective and adverb phrase nodes even when they are unary as well as noun phrases inside PPs and distinguishes between relative and other clauses, which corresponds to Schielen’s

annotation strategies 2 (*Coordinated Categories*), 3 (*Hiding POS tags*) and 5 (*Relative Clauses*). TüBa-D/Z annotation also includes nodes for topological fields (Höhle, 1986) which are not used by Dubey or Schiehlen.[3]

The following changes were made to the base model:

1. *Better handling of unknown words*. The version of BitPar made available by Helmut Schmid only distinguishes between lower- and uppercase words for the purpose of unknown word classification. We modified the program so that it is possible to use a decision tree with regular expressions at the nodes. The current version of the parser uses 42 regular expressions that are used to induce a decision tree with a number of leaves ranging from 79 (vanilla) to 111 (sclass3). In contrast to Schiehlen (2004), we do not use a fully fledged morphological analyzer or other expensive lexical resource.
2. *Verbal complex classification and binarization.* The form of the verb (whether auxiliary, modal or full verb and also whether infinitive, past participle or finite) that is marked in the POS tagset is annotated in the verb form, and subsequent nonfinite verbs are merged in a binary tree with the outmost category reflecting the properties of the whole verbal chunk.
3. *Marking of the complementizer of relative clauses.* The NP or PP containing a relative pronoun or determiner is relabeled from NX to NX-REL (respectively, from PX to PX-REL). This is due to the observation that these NPs usually have a very simple form and that clauses with relative pronouns do not occur outside the complementizer position of a relative clause.
4. *Comparative Phrases.* Like Negra, the TüBa-D/Z annotates comparative phrases (*like Peter*, *as associate chairman*) with the category of the embedded constituent. We move the embedded constituent into a node of its own and rename the node of the comparative phrase, similar to Schiehlen's strategy (11).
5. *Selective Markovization.* While Schiehlen (2004) argues that markovization only helps performance with respect to constituency measures and actually hurts the performance with respect to dependencies, Dubey (2005) gets good results (as measured by constituency measures; unfortunately, there is no dependency-based evaluation of his parser) with a combination of annotating trees and a smoothed second order markovization of the rules. We used markovization in two cases where we knew that the lost horizontal context would not be needed, namely the top-level expansion (from the start symbol to the actual sentence(s) or fragment(s)) and in conjunctions.

[3] Kübler (2005) showed that TüBa-D/Z and Negra behave roughly similarly in terms of parsing performance (in terms of constituent measures, testing on a different part of the same corpus) when considering similar versions of both treebanks (achieved by adding/removing unary nodes and/or topological fields, both being beneficial for the parsing accuracy).

6. *Case marking.* NPs are marked with their case (nominative, accusative, dative, genitive). Determiners are marked with the full morphological information from TüBa-D/Z (including case, number and gender), and pronouns (personal and demonstrative) are marked with case only. This was mainly inspired by the WCDG lexicon, which also has separate entries for each determiner form, whereas the entries for nouns and adjectives are underspecified and grouped by surface form.
7. *Annotation of topological fields.* The MF (Mittelfeld) and VF (Vorfeld) topological fields have been annotated with the arguments contained therein (e.g. MF_ON_OA for a Mittelfeld containing both the subject and the accusative object of a clause). This is to prevent an argument role from being assigned twice.
8. *Annotation of verb valencies.* With the help of the lexicon of the WCDG parser, we annotated verb forms (and the corresponding verb chunks) with approximate valency information. Because the WCDG lexicon often has several alternatives for the valencies of a verb, as well as optional valencies, we shortened the valency descriptions to a string describing just whether arguments of accusative and/or dative case may be expected.[4] Moreover, this information is only annotated for verbs occurring in the corpus and the parser has to guess the valencies of unknown and/or rare verbs.
9. *Classification of clauses and partial clauses.* In the original TüBa-D/Z treebank, only relative clauses are differentiated from other clauses, with no difference between finite, infinitive and other clause complements. The partial clauses that occur in coordinations are also annotated accordingly.
10. *Better information for noun phrases.* In addition to the determiners, we also annotated adjectives (with their endings, which change according to definiteness, gender, number, and case) and nouns (with number and gender). Gender was left underspecified in plural NPs since it is not morphologically marked.
11. *NX/NCX split.* The TüBa-D/Z treebank distinguishes between base NPs and non-base NPs (i.e. with a postmodifying adjunct), as in

    (4) [$_{\text{NX}}$ [$_{\text{NCX}}$ ein Altenheim] [$_{\text{PX}}$ in [$_{\text{NCX}}$ Danzig]]]

    Since this is handled somewhat idiosyncratically in the case of named entities (which are always marked as NCX even when there is an intervening PP postmodification) and differences in the annotation of base vs. non-base NPs can make a noticeable difference (see Bikel, 2004), we first preferred to unify the NX and NCX node labels for parsing (only the '*vanilla*' models have the NX/NCX

[4] The WCDG lexicon contains more extensive information that covers also genitive verb complements (which occur very rarely in normal text) as well as clausal complements (which we decided not to include because the attachment of clausal complements of verbs and nouns is left underspecified in most cases anyway).

split, all others don't). Subsequent experiments (see the rows labeled *sclass*^{2}, *sclass*, *sclass*^{4} in table 3) confirmed Klein and Manning's finding that a model with the NX/NCX split and without insertion of NX nodes above NCX nodes performs best.

12. *MF markovization.* Since the Mittelfeld topological field often contains several noun phrases, we split up the MF nodes after each argument.
13. *NX-MOD split.* In order to distinguish argument NPs from (mostly) temporally modifying NPs, we renamed the latter to NX-MOD.

Of the changes (1)-(3), the decision tree classifier for unknown words has the greatest impact: as noted in table 3, the '*vanilla*' (unmodified treebank grammar), '*vanilla-labels*' (treebank grammar with edge labels appended to the node labels) and '*nocase*' (modifications up to (3)) grammars all show 1-2% improvement over their counterpart without it. The improvement in the case of verb complex binarization is larger when not using the improved unknown word classification, since it also rules out a number of implausible preterminal category assignments.

The difference between the '*nocase*' grammar and the '*simple*' grammar, which includes the case marking improvements up to (7), is easily visible, and ranges from 1.7-3.0% across all text types, with a larger improvement seen for the labeled dependency measures than for unlabeled dependencies.

The annotation of verb valencies and the clause classification bring another improvement of about 2%, as evidenced by the numbers for the '*sclass*' grammar (incorporating the changes up to (10)).

Markovization of the grammar without case information actually hurts performance a little, even in the case of the literary text which has many long NP conjunctions. In the fully annotated version of our grammar, however, leaving out the markovization slightly (but not significantly) deteriorates the results, with a stronger effect on the more difficult text types (serious literature, law text).

The model labeled '*final*' contains all the improvements, the last three giving quite a visible performance improvement on the TüBa-D/Z corpus, and a smaller (about 0.5%) on the other texts.

3 Discussion

In addition to the corpora used by Foth et al. (2004), we used 300 sentences from the TüBa-D/Z corpus that were annotated using the dependency format used for the WCDG parser. The comparison on the TüBa-D/Z section also serves to establish the feasibility of using the automatic dependency conversion for the comparison: when we compare the evaluation results of the parser against the automatically converted treebank with those for the manually annotated sentences, the values are

Corpus		PCFG labeled*	PCFG unlabeled	WCDG labeled**	WCDG unlabeled**
TüBa-D/Z	(1)	81.8	87.2	—	—
	(2)	79.7	85.4	88.1	90.2
Negra		76.9	84.1	87.0	89.0
trivial literature		78.5	86.0	91.1	93.1
serious literature		71.6	80.7	75.4	78.0
online news		76.9	82.0	88.1	89.8
law text		54.2	62.2	86.7	88.8

(*) some labels have been conflated. See the explanation in the text.

(**) the figures on TüBa-D/Z have been determined by a recent version of the WCDG parser, the other figures are taken from (Foth et al., 2004)

(1) against automatically converted dependencies (2) against manual annotation

Table 1: Evaluation results for the PCFG parser

	DET	ATTR	SUBJ	OBJA	SB/OB	PP	ADV	REL
count	536	309	404	186	628	454	414	39
vanilla	0.97	0.92	0.72	0.37	0.61	0.68	0.57	0.14
v/labels	0.97	0.92	0.74	0.43	0.62	0.70	0.58	0.30
simple	0.96	0.92	0.83	0.67	0.75	0.71	0.60	0.25
sclass	0.98	0.94	0.85	0.70	0.78	0.72	0.61	0.33
convert[1]	0.99	0.99	0.99	0.95	0.97	0.89	0.73	0.87
convert[2]	0.99	0.99	0.99	0.95	0.97	0.89	0.73	0.65
convert[3]	0.99	0.99	0.99	0.92	0.96	0.88	0.72	0.38
WCDG	0.98	0.97	0.90	0.80	0.86	0.78	0.79	0.71

[1]: conversion from original treebank, using secondary edges

[2]: conversion from original treebank, ignoring secondary edges

[3]: conversion from original treebank, using relabeling and markovization

ATTR: adjective modifying a noun

SB/OB: subjects and all nominal objects

Table 2: detailed comparison on TüBa-D/Z

similar enough for our purposes (see tables 1 and 2), even with a less-than-perfect annotation conversion (a comparison of the automatic conversion to the manual one yields F=0.92 for all dependencies (labeled), with F=0.97 for verb–nominal argument dependencies, F=0.89 for prepositional phrases and F=72.6 for adverbs only, see table 2). Taking into account the relabeling, the achievable score for a perfect PCFG parse would be at F=0.89 (labeled) or F=0.93 (unlabeled).

Unsurprisingly, the results for the PCFG parser are significantly below those of Foth et al.'s WCDG parser (78.2% vs. 88.1% labeled F-measure on TüBa-D/Z, 76.5% vs 87.0% on Negra), since the latter uses a more detailed lexicon as well as more syntactic features than would be possible to use in a statistical parser without elaborate smoothing. On the other hand, the smaller time consumption of the PCFG parser (about 2 sec./sentence against 68 sec./sentence for the WCDG parser[5]) may recommend its use in contexts where speed is more important than accuracy.

The variability of the results for both parsers is roughly similar, the difference in labeled dependency F-measure ranges around 10-12% for all "normal" texts. The notable exception is the law text, on which the WCDG parser shows no difference to its usual behaviour, but which give the PCFG parser considerable difficulties, probably due to the frequency of constructions that do not normally occur in other written prose. A different case is the "serious literature" text, where both the WCDG and the PCFG parser perform less well than normally, but the difference between both is smaller than expected (only 4% in terms of labeled dependencies) and the PCFG parser even outperforms the WCDG parser in terms of unlabeled dependencies. As our parser is much slower on these sentences than on the others (5sec./sentence versus about 1sec./sentence for the others), it may well be that the higher complexity of the parsing problem is hurting the WCDG parser's performance.

The changes in the annotation yielded a roughly similar improvement over the baseline in terms of absolute F-measure differences (5-7%), but not in terms of error reduction: the improvement on the test portion of TüBa-D/Z (labeled dependencies) gives an error reduction of 21.3%, while that on the law text only gives an error reduction of 10.2%.

These results confirm that firstly, some text types are more difficult to parse than others, and secondly, provided that we do not stray away too far away from the text type we trained the parser on, that both of the parsers we inspected are equally sensitive to text type variation, despite the difference in the approaches to their construction.

[5] a newer version of the WCDG parser uses a statistical shift-reduce parser to get a better first guess, reducing the average parse time to below 10 sec./sentence (Kilian Foth, p.c.)

	SUBJ	OBJA	OBJD	PRED
PCFG parser	86.9	73.5	53.8	62.0
WCDG parser	91.1	82.9	83.0	77.6
(Müller, 2004)	90.8	81.9	64.5	75.6

As in (Müller, 2004), only function labels have been compared, disregarding dependencies. The number reported is the F-measure.

Table 3: Grammatical Function-based evaluation on TüBa-D/Z

	Tüba-D/Z	Negra	online news	trivial literature	serious literature	law text
vanilla[1,3]	80.0	78.0	76.0	81.6	72.5	55.8
v/labels[1,3]	81.2	79.1	77.0	81.0	73.9	57.2
nocase[1]	80.9	79.5	77.3	83.0	74.5	57.1
vanilla[3]	80.9	78.9	77.2	81.6	73.3	56.6
v/labels[3]	82.6	80.1	78.4	81.6	76.7	58.8
v/labels[2,3]	82.4	79.9	77.8	81.8	74.7	57.8
nocase[2]	82.0	80.2	78.3	83.4	75.2	57.4
nocase[2,3]	81.6	80.1	78.4	83.4	74.4	57.7
simple[2]	83.4	81.5	79.8	84.3	77.0	60.5
sclass[2]	84.1	83.4	81.5	85.9	80.3	61.8
sclass[2,3]	84.1	83.3	81.3	85.9	80.0	61.5
sclass	84.8	84.1	81.8	86.0	80.4	61.9
sclass[4]	84.4	83.7	81.7	86.0	80.4	62.1
final	85.4	84.1	82.0	86.0	80.7	62.2

[1]: no decision tree for unknown words [2]: no NX/NCX split [3]: no markovization
[4]: with NX/NCX split and insertion of NX nodes above NCX nodes without an NX parent

Table 4: Influence of the annotation steps: F-measure, unlabeled dependencies

We also tested the parser on sentences 18,603 through 19,602 of the Negra corpus[6], where our parser reaches an F-measure of 83,6%. This is an improvement over the result of Schiehlen (2004), who gets a dependency F-measure of 81,7%, even more so since we used a different corpus as our training set.

If we compare the results of our PCFG parser with either those of the WCDG parser or the parser of Müller (2004), who uses an extended finite state approach (which is radically different from the WCDG parser in terms of the processing approach, but not so much in terms of the information actually used), and knowing that both non-statistical parsers have improved slightly since the results were published, it is quite obvious that there is still vast room for improvement in statistical approaches to parsing German.

Acknowledgements We would like to thank Wolfgang Menzel and Kilian Foth for making available their corpus to us, and Holger Wunsch as well as Sandra Kübler and the three reviewers for providing constructive comments on an earlier version of this paper. The author's work was supported as part of the DFG collaborative research centre (Sonderforschungsbereich) "SFB 441: Linguistische Datenstrukturen".

References

Bikel, D. M. (2004). Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4):479–511.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*.

Daum, M., Foth, K., and Menzel, W. (2004). Automatic transformation of phrase treebanks to dependency trees. In *Proc. 4th Int. Conference on Language Resources and Evaluation (LREC 2004)*.

Dubey, A. (2005). What to do when lexicalization fails: parsing German with suffix analysis and smoothing. In *Proc. 43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*.

Foth, K. (2003). Eine umfassende Dependenzgrammatik des Deutschen. Technical report, Fachbereich Informatik, Universität Hamburg.

[6] we used sentences 1-3000 as development set for Negra

Foth, K., Daum, M., and Menzel, W. (2004). A broad-coverage parser for German based on defeasible constraints. In *KONVENS 2004, Beiträge zur 7. Konferenz zur Verarbeitung natürlicher Sprache*, pages 45–52, Wien.

Gildea, D. (2001). Corpus variation and parser performance. In *EMNLP'2001*.

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

Kübler, S. (2005). How do treebank annotation schemes influence parsing results? or how not to compare apples with oranges. In *Recent Advances of Natural Language Processing (RANLP-2005)*.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proc. 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*.

Müller, F. H. (2004). Annotating grammatical functions in german using finite-state cascades. In *Proc. 20th Int. Conference on Computational Linguistics (COLING 2004)*.

Ratnaparkhhi, A. (1998). *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania.

Schiehlen, M. (2004). Annotation strategies for probabilistic parsing in german. In *Proc. 20th Int. Conference on Computational Linguistics (COLING-2004)*.

Schmid, H. (2004). Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proc. 20th Int. Conference on Computational Linguistics (COLING 2004)*.

Telljohann, H., Hinrichs, E. W., and Kübler, S. (2003). Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Technical report, Seminar für Sprachwissenschaft, Universität Tübingen.