

# Topological Field Parsing of German

**Jackie Chi Kit Cheung**

Department of Computer Science  
University of Toronto  
Toronto, ON, M5S 3G4, Canada  
jcheung@cs.toronto.edu

**Gerald Penn**

Department of Computer Science  
University of Toronto  
Toronto, ON, M5S 3G4, Canada  
gpenn@cs.toronto.edu

## Abstract

Freer-word-order languages such as German exhibit linguistic phenomena that present unique challenges to traditional CFG parsing. Such phenomena produce discontinuous constituents, which are not naturally modelled by projective phrase structure trees. In this paper, we examine *topological field parsing*, a shallow form of parsing which identifies the major sections of a sentence in relation to the clausal main verb and the subordinating heads. We report the results of topological field parsing of German using the unlexicalized, latent variable-based Berkeley parser (Petrov et al., 2006) Without any language- or model-dependent adaptation, we achieve state-of-the-art results on the TüBa-D/Z corpus, and a modified NEGRA corpus that has been automatically annotated with topological fields (Becker and Frank, 2002). We also perform a qualitative error analysis of the parser output, and discuss strategies to further improve the parsing results.

## 1 Introduction

Freer-word-order languages such as German exhibit linguistic phenomena that present unique challenges to traditional CFG parsing. Topic focus ordering and word order constraints that are sensitive to phenomena other than grammatical function produce discontinuous constituents, which are not naturally modelled by projective (i.e., without crossing branches) phrase structure trees. In this paper, we examine topological field parsing, a shallow form of parsing which identifies the major sections of a sentence in relation to the clausal main verb and subordinating heads, when present. We report the results of parsing German using

the unlexicalized, latent variable-based Berkeley parser (Petrov et al., 2006). Without any language- or model-dependent adaptation, we achieve state-of-the-art results on the TüBa-D/Z corpus (Telljohann et al., 2004), with a  $F_1$ -measure of 95.15% using gold POS tags. A further reranking of the parser output based on a constraint involving paired punctuation produces a slight additional performance gain. To facilitate comparison with previous work, we also conducted experiments on a modified NEGRA corpus that has been automatically annotated with topological fields (Becker and Frank, 2002), and found that the Berkeley parser outperforms the method described in that work. Finally, we perform a qualitative error analysis of the parser output on the TüBa-D/Z corpus, and discuss strategies to further improve the parsing results.

German syntax and parsing have been studied using a variety of grammar formalisms. Hockenmaier (2006) has translated the German TIGER corpus (Brants et al., 2002) into a CCG-based treebank to model word order variations in German. Foth et al. (2004) consider a version of dependency grammars known as *weighted constraint dependency grammars* for parsing German sentences. On the NEGRA corpus (Skut et al., 1998), they achieve an accuracy of 89.0% on parsing dependency edges. In Callmeier (2000), a platform for efficient HPSG parsing is developed. This parser is later extended by Frank et al. (2003) with a topological field parser for more efficient parsing of German. The system by Rohrer and Forst (2006) produces LFG parses using a manually designed grammar and a stochastic parse disambiguation process. They test on the TIGER corpus and achieve an  $F_1$ -measure of 84.20%. In Dubey and Keller (2003), PCFG parsing of NEGRA is improved by using sister-head dependencies, which outperforms standard head lexicalization as well as an unlexicalized model. The best

performing model with gold tags achieve an  $F_1$  of 75.60%. Sister-head dependencies are useful in this case because of the flat structure of NEGRA's trees.

In contrast to the deeper approaches to parsing described above, topological field parsing identifies the major sections of a sentence in relation to the clausal main verb and subordinating heads, when present. Like other forms of shallow parsing, topological field parsing is useful as the first stage to further processing and eventual semantic analysis. As mentioned above, the output of a topological field parser is used as a guide to the search space of a HPSG parsing algorithm in Frank et al. (2003). In Neumann et al. (2000), topological field parsing is part of a divide-and-conquer strategy for shallow analysis of German text with the goal of improving an information extraction system.

Existing work in identifying topological fields can be divided into chunkers, which identify the lowest-level non-recursive topological fields, and parsers, which also identify sentence and clausal structure.

Veenstra et al. (2002) compare three approaches to topological field chunking based on finite state transducers, memory-based learning, and PCFGs respectively. It is found that the three techniques perform about equally well, with  $F_1$  of 94.1% using POS tags from the TnT tagger, and 98.4% with gold tags. In Liepert (2003), a topological field chunker is implemented using a multi-class extension to the canonically two-class support vector machine (SVM) machine learning framework. Parameters to the machine learning algorithm are fine-tuned by a genetic search algorithm, with a resulting  $F_1$ -measure of 92.25%. Training the parameters to SVM does not have a large effect on performance, increasing the  $F_1$ -measure in the test set by only 0.11%.

The corpus-based, stochastic topological field parser of Becker and Frank (2002) is based on a standard treebank PCFG model, in which rule probabilities are estimated by frequency counts. This model includes several enhancements, which are also found in the Berkeley parser. First, they use parameterized categories, splitting non-terminals according to linguistically based intuitions, such as splitting different clause types (they do not distinguish different clause types as basic categories, unlike TüBa-D/Z). Second, they take

into account punctuation, which may help identify clause boundaries. They also binarize the very flat topological tree structures, and prune rules that only occur once. They test their parser on a version of the NEGRA corpus, which has been annotated with topological fields using a semi-automatic method.

Ule (2003) proposes a process termed *Directed Treebank Refinement* (DTR). The goal of DTR is to refine a corpus to improve parsing performance. DTR is comparable to the idea of latent variable grammars on which the Berkeley parser is based, in that both consider the observed treebank to be less than ideal and both attempt to refine it by splitting and merging nonterminals. In this work, splitting and merging nonterminals are done by considering the nonterminals' contexts (i.e., their parent nodes) and the distribution of their productions. Unlike in the Berkeley parser, splitting and merging are distinct stages, rather than parts of a single iteration. Multiple splits are found first, then multiple rounds of merging are performed. No smoothing is done. As an evaluation, DTR is applied to topological field parsing of the TüBa-D/Z corpus. We discuss the performance of these topological field parsers in more detail below.

All of the topological parsing proposals predate the advent of the Berkeley parser. The experiments of this paper demonstrate that the Berkeley parser outperforms previous methods, many of which are specialized for the task of topological field chunking or parsing.

## 2 Topological Field Model of German

Topological fields are high-level linear fields in an enclosing syntactic region, such as a clause (Höhle, 1983). These fields may have constraints on the number of words or phrases they contain, and do not necessarily form a semantically coherent constituent. Although it has been argued that a few languages have no word-order constraints whatsoever, most "free word-order" languages (even Warlpiri) have at the very least some sort of sentence- or clause-initial topic field followed by a second position that is occupied by clitics, a finite verb or certain complementizers and subordinating conjunctions. In a few Germanic languages, including German, the topology is far richer than that, serving to identify all of the components of the verbal head of a clause, except for some cases of long-distance dependen-

cies. Topological fields are useful, because while Germanic word order is relatively free with respect to grammatical functions, the order of the topological fields is strict and unvarying.

Type	Fields
VL	(KOORD) (C) (MF) VC (NF)
V1	(KOORD) (LV) LK (MF) (VC) (NF)
V2	(KOORD) (LV) VF LK (MF) (VC) (NF)

Table 1: Topological field model of German. Simplified from TüBa-D/Z corpus’s annotation schema (Telljohann et al., 2006).

In the German topological field model, clauses belong to one of three types: verb-last (VL), verb-second (V2), and verb-first (V1), each with a specific sequence of topological fields (Table 1). VL clauses include finite and non-finite subordinate clauses, V2 sentences are typically declarative sentences and WH-questions in matrix clauses, and V1 sentences include yes-no questions, and certain conditional subordinate clauses. Below, we give brief descriptions of the most common topological fields.

- VF (*Vorfeld* or ‘pre-field’) is the first constituent in sentences of the V2 type. This is often the topic of the sentence, though as an anonymous reviewer pointed out, this position does not correspond to a single function with respect to information structure. (*e.g.*, *the reviewer suggested this case, where VF contains the focus: –Wer kommt zur Party? –Peter kommt zur Party. –Who is coming to the Party? –Peter is coming to the party.*)
- LK (*Linke Klammer* or ‘left bracket’) is the position for finite verbs in V1 and V2 sentences. It is replaced by a complementizer with the field label C in VL sentences.
- MF (*Mittelfeld* or ‘middle field’) is an optional field bounded on the left by LK and on the right by the verbal complex VC or by NF. Most verb arguments, adverbs, and prepositional phrases are found here, unless they have been fronted and put in the VF, or are prosodically heavy and postposed to the NF field.
- VC is the verbal complex field. It includes infinite verbs, as well as finite verbs in VL sentences.

- NF (*Nachfeld* or ‘post-field’) contains prosodically heavy elements such as postposed prepositional phrases or relative clauses.
- KOORD<sup>1</sup> (*Koordinationsfeld* or ‘coordination field’) is a field for clause-level conjunctions.
- LV (*Linksversetzung* or ‘left dislocation’) is used for resumptive constructions involving left dislocation. For a detailed linguistic treatment, see (Frey, 2004).

Exceptions to the topological field model as described above do exist. For instance, parenthetical constructions exist as a mostly syntactically independent clause inside another sentence. In our corpus, they are attached directly underneath a clausal node without any intervening topological field, as in the following example. In this example, the parenthetical construction is highlighted in bold print. Some clause and topological field labels under the NF field are omitted for clarity.

- (1) (a) (*SIMPX* “(VF Man) (LK muß) (VC verstehen) ” , (*SIMPX sagte er*), “ (NF daß diese Minderheiten seit langer Zeit massiv von den Nazis bedroht werden)). ”
- (b) Translation: “One must understand,” **he said**, “that these minorities have been massively threatened by the Nazis for a long time.”

### 3 A Latent Variable Parser

For our experiments, we used the latent variable-based Berkeley parser (Petrov et al., 2006). Latent variable parsing assumes that an observed treebank represents a coarse approximation of an underlying, optimally refined grammar which makes more fine-grained distinctions in the syntactic categories. For example, the noun phrase category *NP* in a treebank could be viewed as a coarse approximation of two noun phrase categories corresponding to subjects and object, *NP^S*, and *NP^VP*.

The Berkeley parser automates the process of finding such distinctions. It starts with a simple binarized X-bar grammar style backbone, and goes through iterations of splitting and merging non-terminals, in order to maximize the likelihood of the training set treebank. In the splitting stage,

<sup>1</sup>The TüBa-D/Z corpus distinguishes coordinating and non-coordinating particles, as well as clausal and field coordination. These distinctions need not concern us for this explanation.

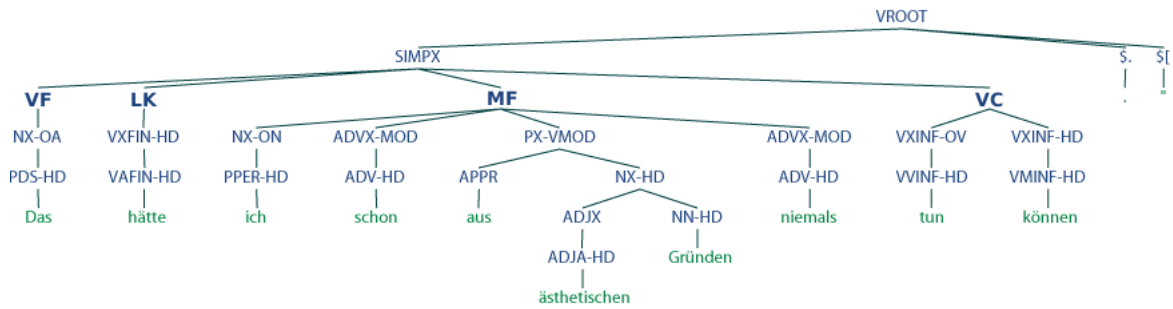


Figure 1: “I could never have done that just for aesthetic reasons.” Sample TüBa-D/Z tree, with topological field annotations and edge labels. Topological field layer in bold.

an Expectation-Maximization algorithm is used to find a good split for each nonterminal. In the merging stage, categories that have been oversplit are merged together to keep the grammar size tractable and reduce sparsity. Finally, a smoothing stage occurs, where the probabilities of rules for each nonterminal are smoothed toward the probabilities of the other nonterminals split from the same syntactic category.

The Berkeley parser has been applied to the TüBaD/Z corpus in the constituent parsing shared task of the ACL-2008 Workshop on Parsing German (Petrov and Klein, 2008), achieving an  $F_1$ -measure of 85.10% and 83.18% with and without gold standard POS tags respectively<sup>2</sup>. We chose the Berkeley parser for topological field parsing because it is known to be robust across languages, and because it is an unlexicalized parser. Lexicalization has been shown to be useful in more general parsing applications due to lexical dependencies in constituent parsing (e.g. (Kübler et al., 2006; Dubey and Keller, 2003) in the case of German). However, topological fields explain a higher level of structure pertaining to clause-level word order, and we hypothesize that lexicalization is unlikely to be helpful.

## 4 Experiments

### 4.1 Data

For our experiments, we primarily used the TüBa-D/Z (Tübinger Baubank des Deutschen / Schriftsprache) corpus, consisting of 26116 sentences (20894 training, 2611 development, 2089 test, with a further 522 sentences held out for future ex-

<sup>2</sup>This evaluation considered grammatical functions as well as the syntactic category.

periments)<sup>3</sup> taken from the German newspaper *die tageszeitung*. The corpus consists of four levels of annotation: clausal, topological, phrasal (other than clausal), and lexical. We define the task of topological field parsing to be recovering the first two levels of annotation, following Ule (2003).

We also tested the parser on a version of the NEGRA corpus derived by Becker and Frank (2002), in which syntax trees have been made projective and topological fields have been automatically added through a series of linguistically informed tree modifications. All internal phrasal structure nodes have also been removed. The corpus consists of 20596 sentences, which we split into subsets of the same size as described by Becker and Frank (2002)<sup>4</sup>. The set of topological fields in this corpus differs slightly from the one used in TüBa-D/Z, making no distinction between clause types, nor consistently marking field or clause conjunctions. Because of the automatic annotation of topological fields, this corpus contains numerous annotation errors. Becker and Frank (2002) manually corrected their test set and evaluated the automatic annotation process, reporting labelled precision and recall of 93.0% and 93.6% compared to their manual annotations. There are also punctuation-related errors, including missing punctuation, sentences ending in commas, and sentences composed of single punctuation marks. We test on this data in order to provide a better comparison with previous work. Although we could have trained the model in Becker and Frank (2002) on the TüBa-D/Z corpus, it would not have

<sup>3</sup>These are the same splits into training, development, and test sets as in the ACL-08 Parsing German workshop. This corpus does not include sentences of length greater than 40.

<sup>4</sup>16476 training sentences, 1000 development, 1058 testing, and 2062 as held-out data. We were unable to obtain the exact subsets used by Becker and Frank (2002). We will discuss the ramifications of this on our evaluation procedure.

<i>Gold tags</i>	<i>Edge labels</i>	<i>LP%</i>	<i>LR%</i>	<i>F<sub>1</sub>%</i>	<i>CB</i>	<i>CB0%</i>	<i>CB ≤ 2%</i>	<i>EXACT%</i>
-	-	93.53	93.17	93.35	0.08	94.59	99.43	79.50
+	-	95.26	95.04	95.15	0.07	95.35	99.52	83.86
-	+	92.38	92.67	92.52	0.11	92.82	99.19	77.79
+	+	92.36	92.60	92.48	0.11	92.82	99.19	77.64

Table 2: Parsing results for topological fields and clausal constituents on the TüBa-D/Z corpus.

been a fair comparison, as the parser depends quite heavily on NEGRA’s annotation scheme. For example, TüBa-D/Z does not contain an equivalent of the modified NEGRA’s parameterized categories; there exist edge labels in TüBaD/Z, but they are used to mark head-dependency relationships, not subtypes of syntactic categories.

## 4.2 Results

We first report the results of our experiments on the TüBa-D/Z corpus. For the TüBa-D/Z corpus, we trained the Berkeley parser using the default parameter settings. The grammar trainer attempts six iterations of splitting, merging, and smoothing before returning the final grammar. Intermediate grammars after each step are also saved. There were training and test sentences without clausal constituents or topological fields, which were ignored by the parser and by the evaluation. As part of our experiment design, we investigated the effect of providing gold POS tags to the parser, and the effect of incorporating edge labels into the nonterminal labels for training and parsing. In all cases, gold annotations which include gold POS tags were used when training the parser.

We report the standard PARSEVAL measures of parser performance in Table 2, obtained by the `evalb` program by Satoshi Sekine and Michael Collins. This table shows the results after five iterations of grammar modification, parameterized over whether we provide gold POS tags for parsing, and edge labels for training and parsing. The number of iterations was determined by experiments on the development set. In the evaluation, we do not consider edge labels in determining correctness, but do consider punctuation, as Ule (2003) did. If we ignore punctuation in our evaluation, we obtain an  $F_1$ -measure of 95.42% on the best model (+ *Gold tags*, - *Edge labels*).

Whether supplying gold POS tags improves performance depends on whether edge labels are considered in the grammar. Without edge labels, gold POS tags improve performance by almost

two points, corresponding to a relative error reduction of 33%. In contrast, performance is negatively affected when edge labels are used and gold POS tags are supplied (i.e., + *Gold tags*, + *Edge labels*), making the performance *worse* than not supplying gold tags. Incorporating edge label information does not appear to improve performance, possibly because it oversplits the initial treebank and interferes with the parser’s ability to determine optimal splits for refining the grammar.

<i>Parser</i>	<i>LP%</i>	<i>LR%</i>	<i>F<sub>1</sub>%</i>
<b>TüBa-D/Z</b>			
This work	95.26	95.04	95.15
Ule	unknown	unknown	91.98
<b>NEGRA - from Becker and Frank (2002)</b>			
BF02 (len. ≤ 40)	92.1	91.6	91.8
<b>NEGRA - our experiments</b>			
This work (len. ≤ 40)	90.74	90.87	90.81
BF02 (len. ≤ 40)	89.54	88.14	88.83
This work (all)	90.29	90.51	90.40
BF02 (all)	89.07	87.80	88.43

Table 3: BF02 = (Becker and Frank, 2002). Parsing results for topological fields and clausal constituents. Results from Ule (2003) and our results were obtained using different training and test sets. The first row of results of Becker and Frank (2002) are from that paper; the rest were obtained by our own experiments using that parser. All results consider punctuation in evaluation.

To facilitate a more direct comparison with previous work, we also performed experiments on the modified NEGRA corpus. In this corpus, topological fields are *parameterized*, meaning that they are labelled with further syntactic and semantic information. For example, VF is split into VF-REL for relative clauses, and VF-TOPIC for those containing topics in a verb-second sentence, among others. All productions in the corpus have also been binarized. Tuning the parameter settings on the development set, we found that parameterized categories, binarization, and including punctuation gave the best  $F_1$  performance. First-order horizontal and zeroth order vertical markoviza-

tion after six iterations of splitting, merging, and smoothing gave the best  $F_1$  result of 91.78%. We parsed the corpus with both the Berkeley parser and the best performing model of Becker and Frank (2002).

The results of these experiments on the test set for sentences of length 40 or less and for all sentences are shown in Table 3. We also show other results from previous work for reference. We find that we achieve results that are better than the model in Becker and Frank (2002) on the test set. The difference is statistically significant ( $p = 0.0029$ , Wilcoxon signed-rank).

The results we obtain using the parser of Becker and Frank (2002) are worse than the results described in that paper. We suggest the following reasons for this discrepancy. While the test set used in the paper was manually corrected for evaluation, we did not correct our test set, because it would be difficult to ensure that we adhered to the same correction guidelines. No details of the correction process were provided in the paper, and descriptive grammars of German provide insufficient guidance on many of the examples in NEGRA on issues such as ellipses, short infinitival clauses, and expanded participial constructions modifying nouns. Also, because we could not obtain the exact sets used for training, development, and testing, we had to recreate the sets by randomly splitting the corpus.

### 4.3 Category Specific Results

We now return to the TüBa-D/Z corpus for a more detailed analysis, and examine the category-specific results for our best performing model (+ *Gold tags*, - *Edge labels*). Overall, Table 4 shows that the best performing topological field categories are those that have constraints on the type of word that is allowed to fill it (finite verbs in LK, verbs in VC, complementizers and subordinating conjunctions in C). VF, in which only one constituent may appear, also performs relatively well. Topological fields that can contain a variable number of heterogeneous constituents, on the other hand, have poorer  $F_1$ -measure results. MF, which is basically defined relative to the positions of fields on either side of it, is parsed several points below LK, C, and VC in accuracy. NF, which contains different kinds of extraposed elements, is parsed at a substantially worse level.

Poorly parsed categories tend to occur infre-

quently, including LV, which marks a rare resumptive construction; FKOORD, which marks topological field coordination; and the discourse marker DM. The other clause-level constituents (PSIMPX for clauses in paratactic constructions, RSIMPX for relative clauses, and SIMPX for other clauses) also perform below average.

<i>Topological Fields</i>				
<i>Category</i>	<i>#</i>	<i>LP%</i>	<i>LR%</i>	<i>F<sub>1</sub>%</i>
PARORD	20	100.00	100.00	100.00
VCE	3	100.00	100.00	100.00
LK	2186	99.68	99.82	99.75
C	642	99.53	98.44	98.98
VC	1777	98.98	98.14	98.56
VF	2044	96.84	97.55	97.20
KOORD	99	96.91	94.95	95.92
MF	2931	94.80	95.19	94.99
NF	643	83.52	81.96	82.73
FKOORD	156	75.16	73.72	74.43
LV	17	10.00	5.88	7.41
<i>Clausal Constituents</i>				
<i>Category</i>	<i>#</i>	<i>LP%</i>	<i>LR%</i>	<i>F<sub>1</sub>%</i>
SIMPX	2839	92.46	91.97	92.21
RSIMPX	225	91.23	92.44	91.83
PSIMPX	6	100.00	66.67	80.00
DM	28	59.26	57.14	58.18

Table 4: Category-specific results using grammar with no edge labels and passing in gold POS tags.

### 4.4 Reranking for Paired Punctuation

While experimenting with the development set of TüBa-D/Z, we noticed that the parser sometimes returns parses, in which paired punctuation (e.g. quotation marks, parentheses, brackets) is not placed in the same clause—a linguistically implausible situation. In these cases, the high-level information provided by the paired punctuation is overridden by the overall likelihood of the parse tree. To rectify this problem, we performed a simple post-hoc reranking of the 50-best parses produced by the best parameter settings (+ *Gold tags*, - *Edge labels*), selecting the first parse that places paired punctuation in the same clause, or returning the best parse if none of the 50 parses satisfy the constraint. This procedure improved the  $F_1$ -measure to 95.24% (LP = 95.39%, LR = 95.09%).

Overall, 38 sentences were parsed with paired punctuation in different clauses, of which 16 were reranked. Of the 38 sentences, reranking improved performance in 12 sentences, did not affect performance in 23 sentences (of which 10 already had a perfect parse), and hurt performance in three sentences. A two-tailed sign test suggests that rerank-

ing improves performance ( $p = 0.0352$ ). We discuss below why sentences with paired punctuation in different clauses can have perfect parse results.

To investigate the upper-bound in performance that this form of reranking is able to achieve, we calculated some statistics on our (+ *Gold tags*, - *Edge labels*) 50-best list. We found that the average rank of the best scoring parse by  $F_1$ -measure is 2.61, and the perfect parse is present for 1649 of the 2088 sentences at an average rank of 1.90. The oracle  $F_1$ -measure is 98.12%, indicating that a more comprehensive reranking procedure might allow further performance gains.

#### 4.5 Qualitative Error Analysis

As a further analysis, we extracted the worst scoring fifty sentences by  $F_1$ -measure from the parsed test set (+ *Gold tags*, - *Edge labels*), and compared them against the gold standard trees, noting the cause of the error. We analyze the parses before reranking, to see how frequently the paired punctuation problem described above severely affects a parse. The major mistakes made by the parser are summarized in Table 5.

<i>Problem</i>	<i>Freq.</i>
Misidentification of Parentheticals	19
Coordination problems	13
Too few SIMPX	10
Paired punctuation problem	9
Other clause boundary errors	7
Other	6
Too many SIMPX	3
Clause type misidentification	2
MF/NF boundary	2
LV	2
VF/MF boundary	2

Table 5: Types and frequency of parser errors in the fifty worst scoring parses by  $F_1$ -measure, using parameters (+ *Gold tags*, - *Edge labels*).

**Misidentification of Parentheticals** Parenthetical constructions do not have any dependencies on the rest of the sentence, and exist as a mostly syntactically independent clause inside another sentence. They can occur at the beginning, end, or in the middle of sentences, and are often set off orthographically by punctuation. The parser has problems identifying parenthetical constructions, often positing a parenthetical construction when that constituent is actually attached to a topological field in a neighbouring clause. The following example shows one such misidentification in

bracket notation. Clause internal topological fields are omitted for clarity.

- (2) (a) TüBa-D/Z: (SIMPX *Weder das Ausmaß der Schönheit noch der frühere oder spätere Zeitpunkt der Geburt macht einen der Zwillinge für eine Mutter mehr oder weniger echt / authentisch / überlegen*).
- (b) Parser: (SIMPX *Weder das Ausmaß der Schönheit noch der frühere oder spätere Zeitpunkt der Geburt macht einen der Zwillinge für eine Mutter mehr oder weniger echt*) **(PARENTHETICAL / authentisch / überlegen.)**
- (c) Translation: “Neither the degree of beauty nor the earlier or later time of birth makes one of the twins any more or less real/authentic/superior to a mother.”

We hypothesized earlier that lexicalization is unlikely to give us much improvement in performance, because topological fields work on a domain that is higher than that of lexical dependencies such as subcategorization frames. However, given the locally independent nature of legitimate parentheticals, a limited form of lexicalization or some other form of stronger contextual information might be needed to improve identification performance.

**Coordination Problems** The second most common type of error involves field and clause coordinations. This category includes missing or incorrect FKOORD fields, and conjunctions of clauses that are misidentified. In the following example, the conjoined MFs and following NF in the correct parse tree are identified as a single long MF.

- (3) (a) TüBa-D/Z: *Auf dem europäischen Kontinent aber hat (FKOORD (MF kein Land und keine Macht ein derartiges Interesse an guten Beziehungen zu Rußland) und (MF auch kein Land solche Erfahrungen im Umgang mit Rußland)) (NF wie Deutschland)*.
- (b) Parser: *Auf dem europäischen Kontinent aber hat (MF kein Land und keine Macht ein derartiges Interesse an guten Beziehungen zu Rußland und auch kein Land solche Erfahrungen im Umgang mit Rußland wie Deutschland)*.
- (c) Translation: “On the European continent, however, no land and no power has such an interest in good relations with Russia (as Germany), and also no land (has) such experience in dealing with Russia as Germany.”

**Other Clause Errors** Other clause-level errors include the parser predicting too few or too many clauses, or misidentifying the clause type. Clauses are sometimes confused with NFs, and there is one case of a relative clause being misidentified as a

main clause with an intransitive verb, as the finite verb appears at the end of the clause in both cases. Some clause errors are tied to incorrect treatment of elliptical constructions, in which an element that is inferable from context is missing.

**Paired Punctuation** Problems with paired punctuation are the fourth most common type of error. Punctuation is often a marker of clause or phrase boundaries. Thus, predicting paired punctuation incorrectly can lead to incorrect parses, as in the following example.

- (4) (a) “*Auch (SIMPX wenn der Krieg heute ein Mobilisierungsfaktor ist) , so Pau , (SIMPX die Leute sehen , daß man für die Arbeit wieder auf die Straße gehen muß) .*”  
 (b) Parser: (SIMPX “(LV Auch (SIMPX wenn der Krieg heute ein Mobilisierungsfaktor ist) ” , so Pau , “ (SIMPX die Leute sehen , daß man für die Arbeit wieder auf die Straße gehen muß) ) . ”  
 (c) Translation: “Even if the war is a factor for mobilization,” said Pau, “the people see, that one must go to the street for employment again.”

Here, the parser predicts a spurious SIMPX clause spanning the text of the entire sentence, but this causes the second pair of quotation marks to be parsed as belonging to two different clauses. The parser also predicts an incorrect LV field. Using the paired punctuation constraint, our reranking procedure was able to correct these errors.

Surprisingly, there are cases in which paired punctuation does not belong inside the same clause in the gold parses. These cases are either extended quotations, in which each of the quotation mark pair occurs in a different sentence altogether, or cases where the second of the quotation mark pair must be positioned outside of other sentence-final punctuation due to orthographic conventions. Sentence-final punctuation is typically placed outside a clause in this version of TüBa-D/Z.

**Other Issues** Other incorrect parses generated by the parser include problems with the infrequently occurring topological fields like LV and DM, inability to determine the boundary between MF and NF in clauses without a VC field separating the two, and misidentifying appositive constructions. Another issue is that although the parser output may disagree with the gold standard tree in TüBa-D/Z, the parser output may be a well-formed topological field parse for the same sentence with a different interpretation, for example because of attachment ambiguity. Each of

the authors independently checked the fifty worst-scoring parses, and determined whether each parse produced by the Berkeley parser could be a well-formed topological parse. Where there was disagreement, we discussed our judgments until we came to a consensus. Of the fifty parses, we determined that nine, or 18%, could be legitimate parses. Another five, or 10%, differ from the gold standard parse only in the placement of punctuation. Thus, the  $F_1$ -measures we presented above may be underestimating the parser’s performance.

## 5 Conclusion and Future Work

In this paper, we examined applying the latent-variable Berkeley parser to the task of topological field parsing of German, which aims to identify the high-level surface structure of sentences. Without any language or model-dependent adaptation, we obtained results which compare favourably to previous work in topological field parsing. We further examined the results of doing a simple reranking process, constraining the output parse to put paired punctuation in the same clause. This reranking was found to result in a minor performance gain.

Overall, the parser performs extremely well in identifying the traditional left and right brackets of the topological field model; that is, the fields C, LK, and VC. The parser achieves basically perfect results on these fields in the TüBa-D/Z corpus, with  $F_1$ -measure scores for each at over 98.5%. These scores are higher than previous work in the simpler task of topological field chunking. The focus of future research should thus be on correctly identifying the infrequently occurring fields and constructions, with parenthetical constructions being a particular concern. Possible avenues of future research include doing a more comprehensive discriminative reranking of the parser output. Incorporating more contextual information might be helpful to identify discourse-related constructions such as parentheses, and the DM and LV topological fields.

## Acknowledgements

We are grateful to Markus Becker, Anette Frank, Sandra Kuebler, and Slav Petrov for their invaluable help in gathering the resources necessary for our experiments. This work is supported in part by the Natural Sciences and Engineering Research Council of Canada.



## References

- M. Becker and A. Frank. 2002. A stochastic topological parser for German. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 71–77.
- S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41.
- U. Callmeier. 2000. PET—a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6(01):99–107.
- A. Dubey and F. Keller. 2003. Probabilistic parsing for German using sister-head dependencies. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 96–103.
- K.A. Foth, M. Daum, and W. Menzel. 2004. A broad-coverage parser for German based on defeasible constraints. *Constraint Solving and Language Processing*.
- A. Frank, M. Becker, B. Crysmann, B. Kiefer, and U. Schaefer. 2003. Integrated shallow and deep parsing: TopP meets HPSG. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 104–111.
- W. Frey. 2004. Notes on the syntax and the pragmatics of German Left Dislocation. In H. Lohnstein and S. Trissler, editors, *The Syntax and Semantics of the Left Periphery*, pages 203–233. Mouton de Gruyter, Berlin.
- J. Hockenmaier. 2006. Creating a CCGbank and a Wide-Coverage CCG Lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 505–512.
- T.N. Höhle. 1983. *Topologische Felder*. Ph.D. thesis, Köln.
- S. Kübler, E.W. Hinrichs, and W. Maier. 2006. Is it really that difficult to parse German? In *Proceedings of EMNLP*.
- M. Liepert. 2003. Topological Fields Chunking for German with SVM's: Optimizing SVM-parameters with GA's. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP), Bulgaria*.
- G. Neumann, C. Braun, and J. Piskorski. 2000. A Divide-and-Conquer Strategy for Shallow Parsing of German Free Texts. In *Proceedings of the sixth conference on Applied natural language processing*, pages 239–246. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.
- S. Petrov and D. Klein. 2008. Parsing German with Latent Variable Grammars. In *Proceedings of the ACL-08: HLT Workshop on Parsing German (PaGe-08)*, pages 33–39.
- S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.
- C. Rohrer and M. Forst. 2006. Improving coverage and parsing quality of a large-scale LFG for German. In *Proceedings of the Language Resources and Evaluation Conference (LREC-2006)*, Genoa, Italy.
- W. Skut, T. Brants, B. Krenn, and H. Uszkoreit. 1998. A Linguistically Interpreted Corpus of German Newspaper Text. *Proceedings of the ESSLLI Workshop on Recent Advances in Corpus Annotation*.
- H. Telljohann, E. Hinrichs, and S. Kübler. 2004. The TüBa-D/Z treebank: Annotating German with a context-free backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2229–2235.
- H. Telljohann, E.W. Hinrichs, S. Kübler, and H. Zinsmeister. 2006. Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). *Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, Germany*.
- T. Ule. 2003. Directed Treebank Refinement for PCFG Parsing. In *Proceedings of Workshop on Treebanks and Linguistic Theories (TLT) 2003*, pages 177–188.
- J. Veenstra, F.H. Müller, and T. Ule. 2002. Topological field chunking for German. In *Proceedings of the Sixth Conference on Natural Language Learning*, pages 56–62.