Annotating Grammatical Functions for German Using Finite-State Cascades

Frank Henrik Müller
SFB 441: Linguistic Data Structures, Universität Tübingen
Nauklerstr. 35
D-72074 Tübingen
Germany,
fhm@sfs.uni-tuebingen.de

Abstract

We describe an incremental parser which annotates grammatical functions in German on top of a shallow annotation structure consisting of chunks, topological fields and clauses. Since grammatical functions in German are strongly associated with case, the assignment and disambiguation of morphological information plays a crucial role as a step towards the annotation of grammatical functions. All components of the parsing system rely on *finite-state* methods to ensure efficient annotation. All stages of the annotation are robust so that they can deal with unexpected input from the source text or failing intermediate annotation components.

1 Introduction

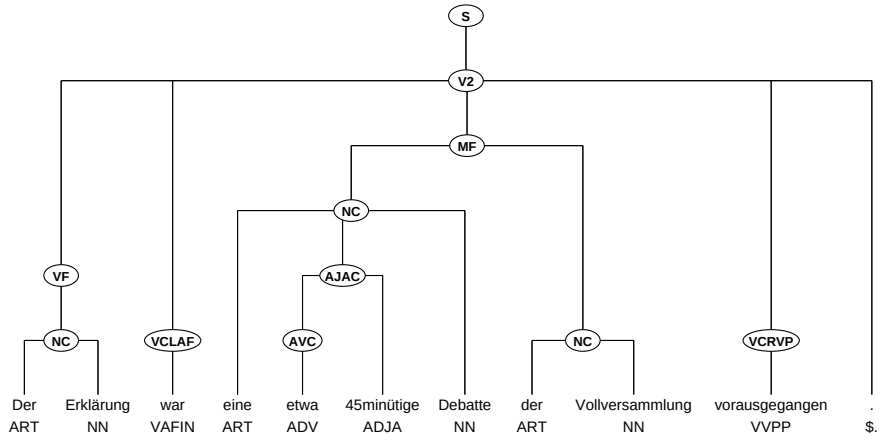
Finite-state (FS) parsing has long been associated with shallow parsing, mainly known under the name of *chunking* or *partial parsing* (Abney, 1996). In this process, non-recursive constituent structures are annotated, typically in an incremental way. These structures may then be used for further processing. Aït-Mokhtar et al. (2002) have shown for French that it is possible to incrementally annotate deeper structures on top of shallow structures using FS methods. We will show that this is feasible for German as well, which differs from French in that it has a far less restricted constituent ordering and in that grammatical functions in German are strongly associated with case. The work presented in this paper is a summary of the work described in Müller (2004).

2 Shallow Structure and Grammatical Functions

Grammatical functions (GFs) and shallow syntactic structures are annotated by two different components because these two linguistic phenomena belong to two different levels of syntactic description. While shallow structures describe constituents which are subject to syntactic restrictions, GFs describe the relations between those constituents. These relations are, in German, often rather subject to lexical selection and manifest themselves in morphological features. While the order of tokens in a chunk is relatively fixed, the order of the GFs is relatively free; and while a noun chunk (NC) is defined purely on the basis of its own inherent features, an NC may **act** as the accusative object (OA) of a certain verb. As regards syntactic annotation, this difference means that shallow structures can be annotated solely using Part-of-Speech (PoS) tags while the annotation of GFs is also in need of a device which assigns and disambiguates morphological information and a lexicon which contains the potential GFs for which a verb or adjective sub-categorizes.

The difference between the two linguistic phenomena can best be illustrated by the sentence in figures 1 and 2. Figure 1 shows the annotation produced by our shallow parsing component KaRoPars (Müller and Ule, 2002); (Ule and Müller, 2004). The tree contains information about the chunk structure, the topological fields (VF=initial field and MF=middle field) and about the borders and the type (V2=verb-second) of the clause. The order of the constituents in this structure is restricted in that the verb chunk containing the finite verb (VCLAF) is always second in V2 clauses and in that the VF is described as the field stretching from the beginning of the sentence to the finite verb, and the MF is defined as the field stretching from the finite verb to the non-finite verbal parts (VCRVP) or to the end of the clause. Word order in chunks is restricted in that e.g. the order of determiner (ART), adjective (ADJA) and noun (NN) is never violated.

This is different for the ordering of GFs, which can be seen in figure 2 showing the same sentence as it is annotated in the treebank TüBa/D-Z (Telljohann et al., 2003), which we



Der Erklärung war eine 45minütige Debatte der Vollversammlung vorausgegangen.
The declaration has a 45-minute debate of the general assembly preceded.
'A 45-minute debate preceded the debate of the general assembly.'

Figure 1: Shallow Annotation Structure

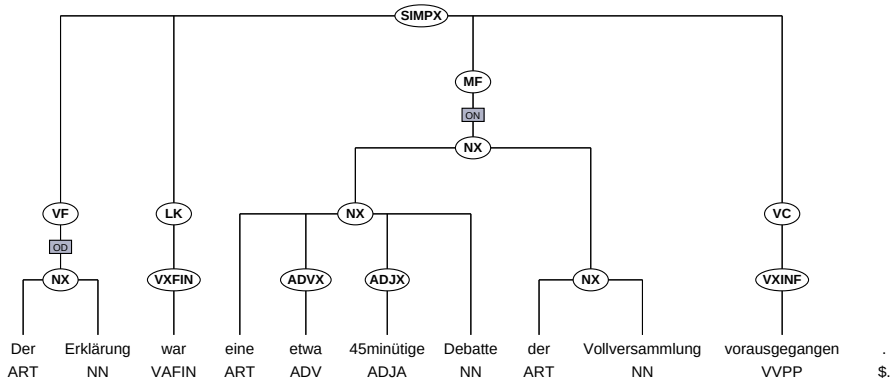


Figure 2: Grammatical Functions Annotation

use as a gold standard for the evaluation of the annotation of grammatical functions.[1] The annotation shows that there is a dative object (OD) which comes first and a nominative object (ON) which comes second. However, the order could just as well be inverse. The distribution of GFs cannot be deduced from the PoS tags and it cannot be deduced from the shallow annotation structure alone, either. The GFs in figure 2 can only be detected by using the sub-categorization (SC) frame of the verb 'vorausgehen' and the morphological features of the relevant chunks.

[1]Phrase-internal head information has been omitted in figure 2.

3 An Incremental Parsing Framework

We perform linguistic annotation incrementally in the sense that different linguistic phenomena are annotated strictly sequentially (cf. figure 3). Following components can use the annotation added by the preceding ones. The first layer of annotation is the layer of PoS tags according to the STTS tagset (Schiller et al., 1995) which may be added by any standard tagger. Then, tokens are assigned lemma information and morphological ambiguity classes – containing *case*, *number*, *gender*, *person* and *inflectional class* information – using the tool DMOR (Schiller, 1995). After this, shallow syntactic structure is annotated by the cascaded robust parser KaRoPars in the order of topo-

logical fields, clauses and chunks – solely making use of the PoS tag information. Afterwards morphological ambiguity is reduced making use of the chunk structure. The following component, which assigns SC frames to the verbs and adjectives, also uses the shallow structure since complex verbs may range over both parts of the sentence bracket, which is part of the topological field annotation. The last step in annotation assigns the GFs, making use of the shallow annotation structure, the reduced morphological ambiguity classes of the chunks and the SC frames of the relevant verbs or adjectives, which we take from IMSLex (Eckle-Kohler, 1999).

4 Annotating Shallow Structures

We define shallow structures as those structures which can be annotated just using PoS tag restrictions without the use of any lexical selection information. For German, shallow structures are, thus, chunks, topological fields and clauses. Chunks are non-recursive kernel phrases (cf. Abney (1996)). Their annotation has been documented best. Topological fields are by now acknowledged as part of a shallow annotation structure which can be integrated into a system of deeper annotation (Neumann et al., 2000); (Hinrichs et al., 2002); (Frank et al., 2003). Topological fields are sections in the German sentence which are described relative to the sentence bracket, which consists of the verbal elements of the sentence and the subordinator in subordinated sentences. The structure of topological fields can be seen as the skeleton of the clause, i.e. it defines the borders of the clause and reveals its type. There are three different types of clauses in German with respect to the position of the finite verb in the topological field structure: verb-first (V1) clauses, verb-second (V2) clauses and verb-last (VL) clauses. The fact that most V1 and V2 clauses are main clauses and most VL clauses are subclauses can be used in the annotation of the structure of complex sentences.

The shallow structure is annotated by cascaded FS transducers. The processing is deterministic in that it invokes a longest-match strategy. A mixed bottom-up and top-down strategy is applied since first topological fields are annotated and, then, clauses. After this, chunks are annotated. The chunking component in itself works top-down. The architecture of the whole shallow parsing component is robust in that, although the following components use the output of the preceding components, they can still be applied if one of the layers of annotation fails. If, for instance, the layers of topological fields and clauses cannot be annotated, then the chunking component is applied to the unchanged output of these components and the following morphological ambiguity reduction component and the GFs annotation component can still be applied.

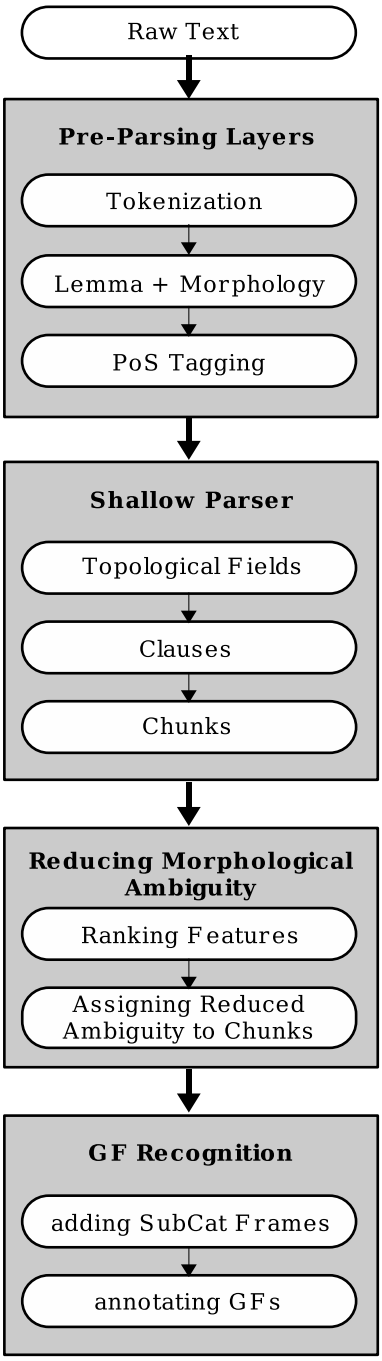


Figure 3: System Architecture

5 Reducing Morphological Ambiguity

Since most of the GFs in German are associated with case, the full morphological disambiguation of a constituent would, in most cases, also reveal which GF it has. In order to detect the morphological information of a constituent, we assign each token in the corpus a morphological ambiguity class using the tool DMOR (Schiller, 1995). Since the tool assigns the morphological features purely on the basis of the token form and the PoS tag independent of any contextual information, the morphological information of each token is highly ambiguous. Although it is not possible to **resolve** ambiguity using local distributional information, it is possible to **reduce** ambiguity by making use of the fact that determiner, adjectives and noun should agree in *case*, *number* and *gender* in chunks. That way, the previously annotated shallow structure can be used to provide contextual information (i.e. chunk boundaries). In order to be better able to handle the morphological information, which consists of more than one attribute-value pair, with an FS approach, we combined the three features *case*, *number* and *gender* into one feature combination (cf. table 1 for an example).

We apply a ranking approach to reduce morphological ambiguity by checking agreement because it is more robust than an alternative intersection approach. In a ranking approach, those feature combinations which occur most often are chosen as the morphological ambiguity class of the chunk (figure 4). In an intersection approach those feature combinations which occur in all relevant tokens are the ambiguity class of the chunk (figure 5). In most cases, the result is the same since the feature combinations occurring in all chunks are also the ones which occur most often. However, there are two possibilities in which this is not the case: First, the morphological annotation tool might have assigned the wrong morphological ambiguity class and, second, one of the relevant tokens might have a typing error, especially in the inflectional morpheme. This is not a rare phenomenon because typing errors in the inflectional morpheme cannot be detected by spelling programs and they easily escape the eye of a (proof) reader.

Figures 4 and 5 show the advantage of a ranking approach over an intersection approach if the wrong morphological ambiguity class is assigned due to a spelling mistake (cf. table 1). The noun chunk (NC) 'einen wichtige Punkt' (*an important point*) has a spelling error in the adjective 'wichtige' which should be spelled 'wichtigen'. Since both forms of the word exist (with different morphological features, however), the misspelled adjective is assigned a morphological ambiguity class which does not comply with the intended function in the chunk.[2] Thus, the input to the morphological ambiguity reduction component is imperfect since it contains two correct and one wrong analysis.

einen	determiner	{asm}
wichtige	adjective	{nsm, nsf, nsn, asf, asn, np0, ap0}
Punkt	noun	{nsm, dsm, asm}

Table 1: Chunk with spelling error in adjective

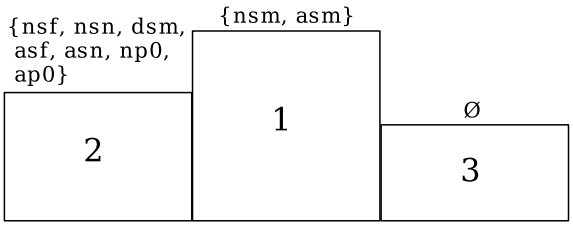


Figure 4: Reducing ambiguity by ranking

The handling of this problem in the two approaches illustrates the advantage of the ranking approach. As figure 5 shows, the intersection approach does not yield any output since the correct morphological *case-number-gender* feature combination 'asm' (for accusative singular masculine) is not element of the intersection of the three morphological ambiguity classes. Figure 4 shows that the ranking approach does yield an output despite the imperfect input. Since none of the feature combinations occurs three times, those which occur twice 'win' and are assigned the chunk as its morphological ambiguity class. In the case at hand, these are 'nsm' and 'asm'. Thus, although, due to the spelling mistake, the ranking approach assigns one incorrect feature combination, it also assigns the correct one. With regard to the concept of robustness, the ranking approach is preferable since, instead of yielding no results, it yields a result at least contain-

[2] In the case of other spelling mistakes, the token might not receive any morphological ambiguity class at all and one could still use a backup component which assigns a default class containing all possible feature combinations.

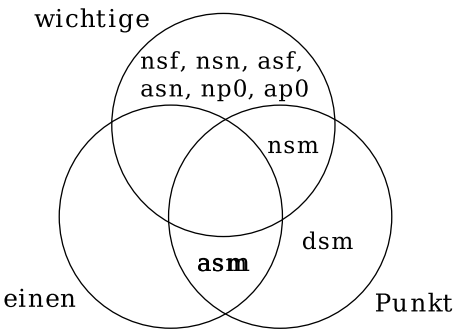


Figure 5: Reducing ambiguity by intersection

ing the correct analysis and thus allows further processing of the output on subsequent stages of annotation which would be blocked otherwise.

We are using an FS approach for the annotation of shallow structures and the annotation of GFs because it is an efficient formalism. There remains, however, the question whether it is possible to model the ranking approach in the FS formalism because ranking involves counting, a process which cannot straightforwardly be modeled in a regular expression grammar. Gerdemann and van Noord (2000) show, however, for the checking of the violation of restrictions in optimality theory (OT) that it is possible to model counting, up to an upper bound, using an FS filter based on matching. Thus, it is possible to use the FS calculus and avoid to use extra-logical procedures in the annotation process.

6 Annotating GFs

GFs can be subdivided along the lines of the distinction between complements, which are typically sub-categorized by a verb or adjective, and adjuncts, which can occur more freely. Since the complements of a verb or an adjective are vital for the structure of the sentence and typically obligatory, we have chosen to concentrate on those phenomena and, for the future, we refer to complements when we speak of GFs. Furthermore, complements also reflect the semantic structure of the sentence according to schemes like 'who did what to whom' and can, thus, be used for applications in information extraction. We annotate the GFs listed in table 2.

The component annotating GFs uses all the linguistic information added by previous components. The chunks are used as targets for the annotation of GFs. Thus, e.g., the chunk 'eine etwa 45minütige Debatte' in figure 1 is assigned the GF nominative object (ON). Attachment ambiguities are not resolved by the GF component. The topological fields structure is used since the ordering in the various fields is subject to ordering preferences which are used for annotation. The clause structure annotated by KaRoPars on the basis of the topological fields structure is used to reduce the search space for potential GFs since the grammatical functions belonging to a certain verb or adjective cannot be outside the clause. This is especially important for complex clauses. With the existing clause structure, each clause can be treated separately, thus using a *divide-and-conquer* strategy. That way, the use of recursive rules, which would be outside the FS formalism, can be avoided.

Since the linear order of GFs in German can vary, from a parsing point of view, the linearization of GFs is ambiguous without a full disambiguation of morphology. Thus, the mechanism dealing with this ambiguity in the annotation of GFs is crucial: The annotation patterns are arranged as a cascade of FS transducers which is applied to the input. After one rule has matched the input string, annotation is effectively blocked. The rules are, thus, applied in a ranked order, and, consequently, the set-up of this order is the decisive feature in the disambiguation mechanism. The ordering of the rules takes advantage of the fact that, although various distributions of GFs are possible, the order is subject to certain principles which can be coded abstractly (Uszkoreit, 1987). However, some of the principles are sometimes conflicting and some of the principles cannot be used without additional semantic information or even world knowledge. Thus, our approach has to resort to those principles which are capable of being integrated into an FS approach. These are the facts that the unmarked order of grammatical functions is: ONs precede accusative objects (OAs) and dative objects (ODs) and: ODs precede OAs if both occur. Furthermore, pronouns precede non-pronouns. Uszkoreit (1987) refers to these rules as linear precedence (LP) rules. We reproduce these rules in figure 6. These rules can be coded as FS automata as regards the LP of different cases and the feature '±PRONOUN'. We do not use the feature 'FOCUS', however, because this would also involve a semantic analysis.

If a verb sub-categorizes for more than one SC frame, then the frame containing most GFs

```
+NOM        <   +DAT
+NOM        <   +ACC
+DAT        <   +ACC
-FOCUS      <   +FOCUS
+PRONOUN    <   -PRONOUN
```

Figure 6: LP rules for grammatical functions taken from Uszkoreit (1987)

```
ON OD OA
ON OA OD
OD ON OA
OA ON OD
OD OA ON
OA OD ON
```

Figure 7: Order of application of rules for SC frame 'ON OD OA'

is applied first in the cascade of FS transducers in order to resolve ambiguity between different SC frames. Within the rules for each SC frame, those rules are applied first which represent the least marked order of GFs. For the SC frame containing ON, OD and OA, the implementation of the LP rules as described in figure 6 is represented in figure 7. The integration of the ±PRONOUN feature into the system did not improve the results of the annotation. The sentence in figure 2, in which the verb subcategorizes for an ON and an OD, can be used to illustrate the process of annotation: In the cascade, first the rule is applied which would assign ON to the first NC and OD to the second. Since, however, the chunk 'der Erklärung' does not contain the feature 'nominative', this rule fails and the inverse order is applied. This rule succeeds and assigns the GFs.

7 Evaluation and Discussion

Table 2 shows the results of the grammatical functions annotation.[3] The categories are those used in TüBa-D/Z. We show precision, recall and $F_{\beta=1}$. The unseen test section for the evaluation consists of the last 1000 sentences of May, 7th, of the corpus. They contain approximately 3000 GFs.[4] Table 2 shows high precision and recall with high-frequency GFs like ON or OA and with PRED. It also shows problems with less-frequent OD and with OPP. Problems with ODs occur because there are so-called free datives which are annotated as OD in our gold standard, TüBa/D-Z, but which are not included in the SC frame of the verb in IMSLex. Problems with OPPs were, to a large extent, caused by the fact that there is no clear–cut distinction between the adverbial and object function of PPs, and TüBa/D-Z and IMSLex differ in their definitions.

Annotating GFs using an FS formalism has also been implemented by Oflazer (2003) for Turkish, by Aït-Mokhtar et al. (2002) for French and by Schiehlen (2003) for German. Like in our approach, Oflazer (2003) annotates complements as GFs. The evaluation is, however, restricted to a 200 sentences corpus and 30 of these sentences are already used for developing the parser. Aït-Mokhtar et al. (2002)'s evaluation is restricted to the two GFs *subject* and *object* while our approach distinguishes six GFs. As there is no distinction between dative and accusative objects in French, this problem does not arise. Schiehlen (2003) also uses an incremental FS approach. However, for the treatment of attachment ambiguities, he uses techniques from constraint-based grammar formalisms in order to deal with underspecification. Thus, Schiehlen (2003) is not purely an FS approach. Schiehlen (2003) evaluates a high number of grammatical functions and other relations. As regards the GFs which are annotated in both Schiehlen (2003) and our approach, our approach is competitive (table 2; Schiehlen (2003) only gives recall). It has, however, to be kept in mind that the test corpus, the evaluation methods and the detailed definition of GFs in Schiehlen (2003) and our approach differ. Furthermore, Schiehlen (2003) annotates dependencies while we annotate GFs as labels as they are coded in TüBa/D-Z. The attachment of the GFs is, however, confined by the topological field and the clause structure since there is typically just one of each GF category per respective subcategorizing lexical item in a clause.

8 Conclusions

We have shown that it is possible to use an FS approach to annotate GFs for German on top of shallow parsing structures. The disambiguation is achieved in two steps: First, a ranking

[3] In order to concentrate on the evaluation of the parser, we use the hand-tagged PoS tags of the TüBa/D-Z. No other gold standard information is used.

[4] The GF component has not yet been optimized for speed, but we expect it to be equally efficient as the shallow parsing component which can be parallelized to annotate more than 1000 tokens per second (cf. Ule and Müller (2004)).

		Precision	Recall	$F_{\beta=1}$	Recall Schiehlen
overall		85.54%	79.65%	82.49	
ON:	Object, Nominative	91.36%	90.20%	90.77	82.6%
OD:	Object, Dative	75.95%	56.07%	64.52	71.9%
OA:	Object, Accusative	81.99%	81.73%	81.86	70.6%
OPP:	Object, prepositional	70.89%	44.94%	55.01	
OS:	Object, Sentence	73.21%	71.93%	72.57	91.2%
PRED:	Predicative	83.50%	76.07%	79.61	55.1%

Table 2: Evaluation of the Grammatical Functions Annotation

approach is used to reduce morphological ambiguity chunk-internally, then SC frames and LP rules for GFs are used to resolve ambiguity. The results of this work compare well with the only other FS approach to GF annotation for German known to us.

9 Acknowledgments

Our thanks go to Tylman Ule for providing the technical framework of KaRoPars which allowed us to add shallow parsing, morphological disambiguation and GFs assignment, to Tylman Ule and Erhard W. Hinrichs for the useful stimulus given to the work presented in this paper, and to the reviewers for their useful comments.

References

Steven Abney. 1996. Partial Parsing via Finite-State Cascades. In *ESSLLI-96 Workshop on "Robust Parsing"*, Prague, Czech Republic.

Salah Aït-Mokhtar, Jean-Pierre Chanod, and Claude Roux. 2002. Robustness beyond shallowness: incremental deep parsing. *Natural Language Engineering*, 8(2–3):121–144.

Judith Eckle-Kohler. 1999. *Linguistisches Wissen zur automatischen Lexikon-Akquisition aus deutschen Textcorpora*. Logos, Berlin.

Anette Frank, Markus Becker, Berthold Crysmann, Bernd Kiefer, and Ulrich Schäfer. 2003. Integrated Shallow and Deep Parsing: TopP meets HPSG. In *ACL 2003*, Sapporo, Japan.

Dale Gerdemann and Gertjan van Noord. 2000. Approximation and Exactness in Finite State Optimality Theory. In *Proceedings the Fifth Workshop of the ACL Special Interest Group in Computational Phonology*, Luxembourg.

Erhard W. Hinrichs, Sandra Kübler, Frank Henrik Müller, and Tylman Ule. 2002. A Hybrid Architecture for Robust Parsing of German. In *LREC 2002*, Las Palmas, Spain.

Frank Henrik Müller and Tylman Ule. 2002. Annotating topological fields and chunks – and revising POS tags at the same time. In *COLING 2002*, Taipei, Taiwan.

Frank Henrik Müller. 2004. *A Finite State Approach to Shallow Parsing and Grammatical Functions Annotation of German*. Ph.D. thesis, Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen. to appear.

Günter Neumann, Christian Braun, and Jakub Piskorski. 2000. A Divide-and-Conquer Strategy for Shallow Parsing of German Free Texts. In *ANLP 2000*, Seattle, WA.

Kemal Oflazer. 2003. Dependency Parsing with an Extended Finite-State Approach. *Computational Linguistics*, 29(4):515–544.

Michael Schiehlen. 2003. Combining Deep and Shallow Approaches in Parsing German. In *ACL 2003*, Sapporo, Japan.

Anne Schiller, Simone Teufel, and Christine Thielen, 1995. *Guidelines für das Taggen deutscher Textcorpora mit STTS*. IMS Stuttgart und SfS Tübingen.

Anne Schiller. 1995. DMOR: Benutzer-Handbuch. Draft, IMS, Universität Stuttgart.

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

Tylman Ule and Frank Henrik Müller. 2004. KaRoPars: Ein System zur linguistischen Annotation großer Text-Korpora des Deutschen. In A. Mehler and H. Lobin, editors, *Automatische Textanalyse. Systeme und Methoden zur Annotation und Analyse natürlichsprachlicher Texte*, pages 185–202, Opladen. VS Verlag.

Hans Uszkoreit. 1987. *Word order and constituent structure in German*, volume 8 of *CSLI lecture notes*. CSLI, Menlo Park, CA.