Verbal Morphosyntactic Disambiguation through Topological Field Recognition in German-Language Law Texts

Kyoko Sugisaki and Stefan Höfler[*]

University of Zurich, Institute of Computational Linguistics,
Binzmühlestrasse 14, 8050 Zürich, Switzerland
{sugisaki,hoefler}@cl.uzh.ch

http://www.cl.uzh.ch

Abstract. The morphosyntactic disambiguation of verbs is a crucial pre-processing step for the syntactic analysis of morphologically rich languages like German and domains with complex clause structures like law texts. This paper explores how much linguistically motivated rules can contribute to the task. It introduces an incremental system of verbal morphosyntactic disambiguation that exploits the concept of topological fields. The system presented is capable of reducing the rate of POS-tagging mistakes from 10.2% to 1.6%. The evaluation shows that this reduction is mostly gained through checking the compatibility of morphosyntactic features within the long-distance syntactic relationships of discontinuous verbal elements. Furthermore, the present study shows that in law texts, the average distance between the left and right bracket of clauses is relatively large (9.5 tokens), and that in this domain, a wide context window is therefore necessary for the morphosyntactic disambiguation of verbs.

Keywords: Morphosyntactic disambiguation, topological field model, Constraint Grammar, law texts, German verbs, POS-tagging

1 Introduction

This paper reports on the development of a rule-based system for the morphosyntactic disambiguation of verbs as a preprocessing component of a supertagger for law texts. The morphosyntactic disambiguation of verbs is a crucial step for recognizing clause structures in a morphologically rich language like German. German verbal complexes are often realised as discontinuous constituents. Moreover, German verbal morphology exhibits some degree of syncretism: verbal inflectional forms and morphosyntactic features are not always in one-to-one

[*] This project was funded under Swiss National Science Foundation grant Topological4701.

relationships. Especially for the legislative domain, the morphosyntactic disambiguation of verbs is a challenging task since clausal structures in law texts are particularly complex. Due to the frequency of verb phrase coordinations and embedded clauses (cf. [8, 17]), the distances between the heads of clauses (e.g. finite verbs and complementisers) and their verbal complements are often relatively long and intricate.

In this paper, we present a rule-based system for morphosyntactic disambiguation of verbs that exploits the concept of topological fields, and we explore to what degree our linguistically motivated rule-based system can resolve verbal morphosyntactic ambiguities in law texts.

The paper is organised as follows. In the next section, we describe the general architecture of our supertagger. In section 3, we present the two major components of verbal morphosyntactic disambiguation. In section 4, we evaluate the performance of our system and discuss the rate of the reduction in part-of-speech tagging errors.

2 Overview: Supertagger

We have been developing a supertagger for the syntactic analysis of Swiss law texts written in German. Suppertagging is an “almost parsing” approach in the sense that the supertags represent rich syntactic information such as valence, voice and grammatical functions [5, 9, 15] and a parser needs then “only combine the individual supertaggs” [1]. Our supertagger is part of a project aimed at detecting style guide violations in legislative drafts [12]. To detect stylistically undesirable syntactic constructions, our supertagger aims at tagging core syntactic structures such as topological fields and grammatical functions. It consists of a pipeline with the following components:

1. Sentence segmentation and tokenization
2. Morphological analysis
3. Morphosyntactic disambiguation of verbs
4. Morphosyntactic disambiguation of nouns
5. Grammatical function recognition

Sentence segmentation and tokenization (component 1) are carried out as described in [12].

For the morphological analysis (component 2), our system employs Gertwol [7]. Gertwol is a classical two-level rule-based morphological analyser and provides fine-grained morphosyntactic features. However, Gertwol does not provide any analysis if it cannot not find the root of a word in its lexicon. In these cases, the system uses the analysis of the statistical decision-tree-based POS-tagger TreeTagger [19] to complete the output of Gertwol: the system identifies the set of possible morphosyntactic features on the basis of the inflectional endings of the tokens unknown to Gertwol and the POS-tags that TreeTagger returns for them. If a token has, for example, the ending *-en* and is analysed as an infinite verb by TreeTagger, two possible morphosyntactic feature sets, that for verbs in

Table 1. Exemplification of the topological field model: occupation of the left and right brackets in the templates of the three clause types as found in sentence (1).

Vorfeld	Left Bracket (LB)	Mittelfeld	Right Bracket (RB)	Nachfeld
Verb-first clause (V1): LB = finite verb, RB = verb complements				
	Stellt	*die Zollverwaltung Unregelmässigkeiten*	*fest,*	
Verb-second clause (V2): LB = finite verb, RB = verb complements				
so	*verweigert*	*sie den Abschluss des Transitverfahrens*		
[und]	*hält*	*die Sicherheit*	*zurück*	
Verb-final clause (VL): LB = subord. conj. / compl., RB = verb complex				
	bis	*die mit bedingter Zahlungspflicht veranlagten Einfuhrzollabgaben*	*bezahlt sind.*	

3rd person plural indicative and that for infinitives, are generated. TreeTagger has proven to be robust and its performance with regard to unknown words is relatively high [21].

The three main components of the system, dedicated to the morphosyntactic disambiguation of verbs (component 3), the morphosyntactic disambiguation of nouns (component 4) and the recognition of grammatical functions (component 5), respectively, have been implemented in the framework of Constraint Grammar. Constraint Grammar [13] is a grammar formalism that has been successfully employed for tasks such as English POS-tagging [22] or NP chunking [23]. We employ VISLCG2[1] to compile hand-crafted Constraint Grammar rules.

In the remainder of this paper, we will focus on component 3 and its strategies for the morphosyntactic disambiguation of verbs.

3 Verbal Morphosyntactic Disambiguation through Topological Field Recognition

The morphosyntactic disambiguation of German verbal elements is a challenging task: German verb forms are morphosyntactically highly ambiguous as syncretism is very common in German verb paradigms. The inflectional ending -*en*, for example, is used to mark 1st person plural (e.g. *wir trink-en* 'we drink'), 3rd person plural (e.g. *sie trink-en* 'they drink') and infinitive (e.g. *trink-en* 'to drink'). On top of that, in tenses other than present and preterite, verbal morphosyntactic properties such as mood and diathesis are realised via periphrasis (i.e. multiword expressions). Depending on the clause type in which they occur, these periphrases appear as continuous or discontinuous constituents.

[1] `http://beta.visl.sdu.dk/` (last visited on 15/05/2013)

3.1 The Topological Field Model

Traditionally, German clause structure has been described in terms of topological fields [4, 14]. The topological fields of a clause are the different positions in which non-verbal constituents can appear: the vorfeld, the mittelfeld and the nachfeld. They are defined relative to the positions in which the heads of the clause (e.g. finite verbs and complementisers) and their verbal complements (e.g. infinitives, participles and separable verb prefixes) can be placed: the left and right bracket of the clause, respectively (cf. Table 1).

Depending on the position of the verbal elements in a clause, the topological field model distinguishes three types (or templates) of German clauses with a different template each: verb-first clauses (V1), verb-second clauses (V2) and verb-final clauses (VF) [3, pp. 864ff]. Table 1 illustrates how the following example sentence is analysed according to this distinction:

(1) Stellt die Zollverwaltung Unregelmässigkeiten fest, so verweigert sie den Abschluss des Transitverfahrens und hält die Sicherheit zurück, bis die mit bedingter Zahlungspflicht veranlagten Einfuhrzollabgaben bezahlt sind.[2]

‘If the customs administration recognises irregularities, it refuses the completion of the transit procedure and retains the security until the import customs fees rated with conditioned duty of payment have been paid.’

Depending on the clause type, different elements can occupy the left and right bracket of a German clause. The left bracket of verb-first clauses (imperative sentences, interrogative sentences, certain conditional clauses) and verb-second clauses is occupied by the finite verb. The right bracket is filled by verbal complements such as separable verb prefixes and, where the finite verb is an auxiliary or a modal, infinitives and participles. In contrast, the left bracket of verb-final clauses (most types of subordinate clauses) is occupied by a subordinating conjunction or a complementiser, whereas the whole verbal complex of these clauses appears in the right bracket.[3] The verbal complex is thus a continuous element in verb-final clauses but can be realised as a discontinuous periphrasis in verb-first and verb-second clauses.

3.2 Approach

Taking into account the language-specific morphosyntactic configurations mentioned above, we propose a verbal disambiguation system for German that is based on the topological field model. The topological field model was first employed for the identification of clause boundaries by Neumann et al. [16]; since,

[2] Art. 155 para. 2 Customs Ordinance (SR *631.01*)

[3] In the present study, relative pronouns have also been considered to occupy the left bracket, although, from a theoretical perspective, they actually appear in vorfeld position. For practical reasons, this simplification seemed justifiable as, in standard German, the left bracket of relative clauses always remains empty.

Table 2. A selection of the heuristics used by the system and the hard topological-field rules on which they are based. (For exhibitory purposes, some of the heuristics are rendered in a slightly simplified form.)

Nr.	**Rule:** Heuristic
General	
G1	**A past participle requires an auxiliary verb:** If a potential past particle is not preceded or immediately followed by an auxiliary verb within the same sentence, then discard the features PART PERF.
...	...
Left Bracket	
L1	**The left bracket of V1 clauses is a single finite verb:** If a verb appears in sentence-initial position, select the feature FINITE from its set of possible features, mark it as left bracket and identify the clause type as V1.
L2	**The left bracket of V2 clauses is a single finite verb:** If a verb in sentence-internal position is not preceded by an auxiliary or modal in the left bracket of a V1-clause, select the feature FINITE from its set of possible features, mark it as left bracket and identify the clause type as V2.
L3	**The left bracket of V1 and V2 clauses is a single finite verb:** If a modal verb is not adjacent to other verbal elements, select the feature FINITE from its set of possible features, mark it as left bracket.
L4	**The left bracket of VF clauses is a conjunction or a complementiser:** If a potential conjunction is indirectly followed by a finite verb and a punctuation mark or a coordinating conjunction, then mark it as left bracket and identify the clause of VF.
...	...
Right Bracket	
R1	**A modal verb requires an infinitive:** If a potential infinitive is preceded by a modal verb at the left-bracket position of a V1 or V2 clause, then select its feature INFINITIVE and mark it as right bracket.
R2	**The auxiliary *werden* requires an infinitive for future tense:** If a potential infinitive is preceded by *werden* at the left-bracket position of a V1 or V2 clause, then select its feature INFINITIVE and mark it as right bracket.
R3	**The auxiliary *haben* requires an infinitive for perfect tense:** If a potential infinitive is preceded by *haben* at the left-bracket position of a V1 or V2 clause, then select its feature INFINITIVE and mark it as right bracket.
R4	**The auxiliaries *werden/sein* require a past particple for passive voice:** If a potential past participle is preceded by *werden* or *sein* at the left-bracket position of a V1 or V2 clause, then select its feature PAST PARTICIPLE and mark it as right bracket.
R5	**The right bracket of VF clauses contains a finite verb:** If a verb is directly followed by a punctuation mark or a coordinating conjunction and preceded by the left bracket of a VF-clause, then select its feature FINITE and mark it as right bracket.
R6	**Lexical verbs can have a separable verb prefix:** If a potential verb prefix is directly followed by a punctuation mark or a coordinating conjunction and preceded by a lexical verb at left-bracket position, then select its feature VERB PREFIX and mark it as right bracket.
...	...

Table 3. Incremental morphosyntactic disambiguation of elements at brackets in sentence (1).

	Stellt	*fest*	*verweigert*	*hält*	*zurück*	*bis*	*bezahlt*	*sind*
Input: Gertwol	PL2 PL2 SG3 PP	ADJ PREF	SG3 PP PL2 PL2 PL2	SG3	PREF ADV	CONJ PREP ADV	SG3 PL2 PL2 PP	PL1 PL3
Step 1a: Morphosynt. disambiguation	SG3		SG3	SG3		CONJ		
Step 1b: Topological field recogn.	LB-V1		LB-V2	LB-V2		LB-VF		
Step 2a: Right brackets disambiguation		PREF			PREF		PP	PL3
Step 2b: Right brackets labeling		RB-V1			RB-V2		RB-VF	RB-VF

it has also been applied in the pre-processing routines of deep syntactic parsers [2, 6, 10]. In our system, it is used for defining rules for verbal morphosyntactic disambiguation. Table 2 shows a selection of the heuristics used by our system and the syntactic rules on which they are based.

Our system proceeds in two steps: in a first step, it disambiguates verbal elements in left-bracket position and determines the clause type, and in a second step, it disambiguates verbal elements in right-bracket position. The second step depends on the completion of the first step as heuristics for right-bracket elements frequently build on knowledge about left-bracket elements (cf. Table 2, rules R1ff.): the morphosyntactic features of verbal elements in right-bracket position are disambiguated by checking the compatibility of their features with those of the corresponding left-bracket elements. The details of what is being checked fall from the morphosyntactic properties of the predicate as a whole (e.g. mode, tense, diatheses) and the type of the clause.

In each step, the heuristics exemplified in Table 2 are applied in a specific order. The order is relevant as some heuristics build on the output of other heuristics. An example is Rule L2, which is concerned with detecting left brackets of verb-second clauses and disambiguating the morphosyntactic features of the corresponding verb form: it exploits information that has previously been added by Rule L1, namely information on the presence of the left bracket of a verb-first clause in the respective context. Morphosyntactic disambiguation thus happens incrementally not just between the two steps but also within.

3.3 Step-by-Step Example

In what follows, we illustrate the two-step procedure of our system by tracking how it processes the aforementioned sentence (1), which we repeat in (2):

(2) Stellt die Zollverwaltung Unregelmässigkeiten fest, so verweigert sie den Abschluss des Transitverfahrens und hält die Sicherheit zurück, bis die mit bedingter Zahlungspflicht veranlagten Einfuhrzollabgaben bezahlt sind.[4]

‘If the customs administration recognises irregularities, it refuses the completion of the transit procedure and retains the security until the import customs fees rated with conditioned duty of payment have been paid.’

Table 3 gives an overview of the morphosyntactic analyses Gertwol returns for each bracket candidate contained in the sentence, i.e. for each token that is a potential left or right bracket (Input), and it illustrates how these analyses are gradualy disambiguated in the processing steps performed by our system (Steps 1a–2b).

Step 1: Left bracket detection and disambiguation

Step 1 is concerned with detecting word forms that serve as left brackets and with determinating the clause type. At the same time, the morphosyntactic analyses of word forms identified as left-bracket elements are disambiguated.

The first left-bracket candidate encountered by the system is the verb form *stellt.* Gertwol yields the following possible morphosyntactic analyses for this token:[5]

(3)
```
"stellt"
      "stell~en"   "V IND PRÄS PL2"
      "stell~en"   "V IMP PRÄS PL2"
      "stell~en"   "V IND PRÄS SG3"
      "stell~en"   "V PART PERF"
```

The system applies a domain-specific heuristic and discards these two analysis because, in general, there are no second-person statements in legislative texts.

The third analysis identifies the word form as a third-person singular verb (V SG3); the fourth analysis interprets it as a past participle (PART PERF). The fourth analysis is discarded because past participles in sentence-initial position are always followed by an auxiliary verb (e.g. *Gekauft habe ich aber dann doch das billigere Auto*), which is not the case in the present sentence. The only

[4] Art. 155 para. 2 Customs Ordinance (SR *631.01*)

[5] To keep the morphosyntactic features of verbs unique per token, redundant features generated by Gertwol are deleted. Tags: V = verb, IND = indicative, PRÄS = present, PL2 = 2nd person plural, IMP = imperative, SG3 = 3rd person singular, PART = participle, PERF = perfect.

remaining analysis is thus the one that interprets the word form in question as a third-person singular verb in present tense indicative.

Given the constraints described by the topological field model (cf. Table 1), the fact that a finite verb occurs in sentence-initial position means that the respective token is the left-bracket of a verb-first clause (cf. Rule L1 in Table 2). The system thus labels the token *stellen* accordingly (LB-V1).

The next left-bracket candidate to be considered by the system is *verweigert*. Gertwol returns the following five morphosyntactic analyses for this token:[6]

```
(4)    "verweigert"
           "ver|weig~er~n"   "V IND PRÄS SG3"
           "ver|weig~er~n"   "V PART PERF"
           "ver|weig~er~n"   "V IND PRÄS PL2"
           "ver|weig~er~n"   "V KONJ PRÄS PL2"
           "ver|weig~er~n"   "V IMP PRÄS PL2"
```

Once more, the system discards all analyses that identify the token as a second-person verb (i.e. the last three analyses listed) as legislative texts generally do not contain second-person statements.

The second analysis listed, containing the feature combination PART PERF, is also discarded by the system: if *verweigert* was a past participle, it would have to be either preceded or immediately followed by an auxiliary verb (Rule G1).

The token *verweigert* has thus been morphosyntactically disambiguated as a third-person singular verb in present indicative. The fact that it is a finite verb and that it is preceded (a) by a verb-first clause and (b) by a comma followed by the adverb *so*, furthermore indicates that *verweigert* is the left bracket of a verb-second clause; The system labels it accordingly.

In a similar fashion, the following two left-bracket candidates, *hält* and *bis*, are identified as the left bracket of a verb-second clause and a verb-final clause, respectively, while the final two candidates, *bezahlt* and *sind*, are identified as not being left brackets (cf. Table 3).

Step 2: Right bracket disambiguation and labeling

Step 2 is concerned with detecting right brackets; at the same time, the morphosyntactic analyses of the respective word forms are disambiguated. Specifically, the system detects and disambiguates tokens that serve as right brackets by checking the compatibility of their morphosyntactic features with those of the left brackets preceding them.

The first right-bracket candidate encountered by the system is the token *fest*. Morphosyntactically, *fest* can either be a predicative adjective or a separable verb prefix. However, only the latter analysis is compatible with the lexical verb in preceding left bracket (*stellt*); The system thus discards the former analysis and tags the token as the right bracket of the respective verb-first clause

[6] KONJ = conjunctive

Table 4. Performance of the system at detecting and disambiguating bracket candidates: Recall.

	correct	wrong	total
TreeTagger	281 tokens (89.8%)	32 tokens (10.2%)	313 tokens (100.0%)
Our system	308 tokens (98.4%)	5 tokens (1.6%)	313 tokens (100.0%)

(Rule R6 in Table 2). By applying the same rule, the next candidate, *zurück*, is disambiguated and identified as the right bracket belonging to the verb-second clause with the finite verb *hält*. The remaining two candidates, *bezahlt* and *sind*, are disambiguated and identified as right brackets by applying Rules G1 and R5, respectively.

4 Evaluation

The strategies for verbal morphosyntactic disambiguation and topological field recognition presented in the previous section have been evaluated over 100 sentences (2,370 tokens) that were randomly selected from the the Swiss Legislation Corpus [11].

4.1 Verbal Morphosyntactic Disambiguation

To evaluate the performance of our verbal morphosyntactic disambiguation system against a gold standard, we manually annotated all potential left- and right-bracket elements (bracket candidates, i.e. potential verbal elements, subordinating conjunctions, complementisers, relative pronouns) in the test sentences. We then processed the same test sentences with our system and compared its automatic annotations with those provided by TreeTagger. To be able to compare the output of the two systems, we converted our Gertwol-based output into the Stuttgart-Tübingen Tagset (STTS) [18] used by TreeTagger.

As shown in Table 4, 308 of the 313 tokens that were tagged in the gold standard were analysed correctly by our system; Our system had a *recall* of 98.4%. In comparison, TreeTagger only achieved a recall of 89.8%. The results of our system thus constitute an improvement of 8.6% from those obtained by TreeTagger.

30 of the 32 tokens wrongly analysed by TreeTagger (i.e. 93.8%) were correctly analysed by our system. Our system mainly proved superior to TreeTagger at tagging right-bracket candidates. Right-bracket candidates are always verbal elements, and verbal elements generally exhibit a relatively high degree of morphosyntactic ambiguity: on average, Gertwol returned 3.3 analyses per token for the verbal elements in our test data. Consequently, all tokens wrongly analysed by TreeTagger were morphologically ambiguous verb forms, e.g. verb forms with the inflectional endings *-en* or *-t*.

The most frequent type that TreeTagger failed to analyse correctly were finite verbs ending in *-en* that appeared in the right bracket of a verb-final clause (9 of 32 tokens). TreeTagger wrongly interpreted these verb forms as infinitives.

Table 5. Performance of the system at detecting and disambiguating bracket candidates: Precision.

	correct	wrong	total
TreeTagger	281 tokens (98.3%)	5 tokens (1.7%)	286 tokens (100.0%)
Our system	308 tokens (99.7%)	1 tokens (0.3%)	309 tokens (100.0%)

To correctly disambiguate right-bracket candidates, information about the corresponding left-bracket elements is required. Our system performed better at the task precisely because it has access to such information. In contrast, the context window used by TreeTagger and other n-gram-based taggers does not seem to be wide enough for domains with relatively complex clause structures such as law texts. Indeed, we found that in the sentences we used for the evaluation, the distance between the left and right bracket amounted to a comparatively high average of 9.54 tokens.

An additional but related explanation of why our system performed better than TreeTagger arises from the fact that some of the tokens for which TreeTagger returned wrong analyses occurred in syntactic structures that are frequent in law texts but not in the newspaper texts TreeTagger was trained on (e.g. verb-first clauses and adverbial participle phrases with participle inversion).

There were also three tokens that were wrongly analysed by our system but correctly analysed by TreeTagger. These errors in the output of our system were caused (a) by the correct analysis not being included in the output provided by Gertwol (*aufrecht* not analysed as prefix), (b) by our system wrongly interpreting a definite article as a relative pronoun, and (c) by a specific syntactic structure not yet taken into account in the disambiguation rules (extraposition of a prepositional phrase within a relative clause in the vorfeld of a verb-second clause).

Our system achieved a *precision* of 99.7%. As shown in Table 5, 308 of the 309 tokens tagged by our system were analysed correctly. TreeTagger achieved a slightly lower precision of 98.3%.

The one incorrect analysis returned by our system was a relative pronoun erroneously tagged as the definite article of a participle phrase. In comparison, TreeTagger misinterpreted three relative pronouns as definite articles; another two mistakes were caused by the phrases *wie folgt* (‘as follows’) and *von sich aus* (‘on its own’).

In summary, TreeTagger achieved an F1 score of 93.8% while our system achieved an F1 score of 99.0%. These results indicate that rule-based morphosyntactic disambiguation can indeed substantially improve the performance of a part-of-seech tagger.

4.2 Topological Field Labeling

We have also used the test data described above to evaluate the performance of our system with regard to recognising topological fields by determining the left

Table 6. Performance of the topological field labeling system.

Recall	Precision	F1-score
95.3% (*286/300 tokens*)	99.7% (*286/287 tokens*)	97.4%

and right brackets of clauses. To this aim, we manually annotated the left and right brackets (300 tokens in total) contained in the sentences selected from the corpus. As shown in Table 6, our system correctly detected 95.3% (286 tokens) of all brackets (recall), and 99.7% (286 tokens) of the tokens that our system marked as brackets (287 tokens) had been identified correctly (precision). In sum, our system thus achieved an F1-score of 97.4% at the task of recognising left and right brackets.

Of the 15 errors (14 false negatives and 1 false positive) that occurred, 12 were the direct or indirect result of a wrong morphosyntactic disambiguation: failure to detect a left bracket (e.g. because a subordinating conjunction had been wrongly analysed as an adverb) frequently also lead to a failure to detect the corresponding right bracket (e.g. because the next finite verb would then be correctly identified as a right-bracket element).

5 Conclusion

The morphosyntactic disambiguation of verbs is a crucial pre-processing step for the syntactic analysis of morphologically rich languages like German and domains with complex clause structures like law texts. In this paper, we explored how much linguistically motivated rules can contribute to the task. We presented an incremental system of verbal morphosyntactic disambiguation that exploits the concept of topological fields. In the evaluated sentences extracted from a corpus of German-language law texts, our system achieved a F1 score of 99.0%

The system proved to be capable of reducing the rate of POS-tagging mistakes from 10.2% in a state-of-the-art statistical tagger to 1.6%. Our evaluation showed that this reduction was mostly gained through checking the compatibility of morphosyntactic features within the long-distance syntactic relationships of discontinuous verbal elements in the left and right brackets of clauses. The present study also showed that in law texts, the average distance between the left and right bracket of clauses is relatively large (9.5 tokens), and that in this domain, a wide context window is therefore necessary for the morphosyntactic disambiguation of verbs.

The present study suggests that such a rule-based system, if employed as a post-processing component, may be able to make a significant contribution to improving the quality of POS-tagging, especially in long-distance discontinuous verbal periphrases in German.

In the future, we plan to use information on the left and right brackets of clauses as additional input for determining grammatical functions.

References

1. Bangalore, S., Joshi, A. K.: Supertagging: an approach to almost parsing. Computational Linguistics, 25(2) (1999)
2. Becker, M., Frank. A.: A Stochastic Topological Parser for German. In: Proceedings of COLING 2002, pp. 71–77. Association of Computational Linguistics, New York (2002)
3. Dudenredaktion(eds.): Duden - die Grammatik: unentbehrlich für richtiges Deutsch, Duden, vol. 4. Dudenverlag, Mannheim (2009)
4. Dürscheid, C.: Syntax: Grundlagen und Theorien. Vandenhoeck & Ruprecht, Göttingen (2012)
5. Foth, K., By, T., Menzel, W.: Guiding a constraint dependency parser with supertags. In: S. Bangalore and A. K. Joshi. (eds.) Supertagging: Using Complex Lexical Descriptions in Natural Language Processing. MIT Press, Cambridge and London (2010)
6. Frank, A., Becker, M., Crysmann, B., Kiefer, B., Schäfer, U.: Integrated Shallow and Deep Parsing: TopP Meets HPSG. In: Proceedings of ACL-2003, pp. 104–111. Association for Computational Linguistics, New York (2003)
7. Haapalainen, M., Majorin, A.: GERTWOL: ein System zur automatischen Wortformerkennung deutscher Wörter. Technical report, Lingsoft (1994)
8. Hansen-Schirra, S., Neumann, S.: Linguistische Verständlichmachung in der juristischen Realität. In: K. D. Lerch. (eds.) Recht verstehen: Verständlichkeit, Missverständlichkeit und Unverständlichkeit von Recht, Die Sprache des Rechts, vol. 1. Walter de Gruyter, Berlin (2004)
9. Harper, M. P., Wang, W.: Constraint dependency grammars: Superarvs, language modeling, and parsing. In: S. Bangalore and A. K. Joshi. (eds.) Supertagging: Using Complex Lexical Descriptions in Natural Language Processing. MIT Press, Cambridge and London (2010)
10. Hinrichs, E. W., Kübler, S., Müller, F. H., Ule, T.: A hybrid architecture for robust parsing of German. In Proceedings of the 3rd International Confererence on Language Resources and Evaluation (LREC 2002), Las Palmas, Gran Canaria (2002)
11. Höfler, S., Piotrowski, M.: Building Corpora for the Philological Study of Swiss Legal Texts. Journal for Language Technology and Computational Linguistics (JLCL), 26(2), 77–89 (2011)
12. Höfler, S., Sugisaki, K.: From Drafting to Error Detection: Automating Style Checking for Legislative Texts. In: EACL 2012 Workshop on Computational Linguistics and Writing, pp. 9–18. Association for Computational Linguistics, New York (2012)
13. Karlsson, F., Voutilainen, A., Heikkilä, J., Anttila, A.(eds.): Constraint Grammar: A Language- Independent System for Parsing Unrestricted Text. Mouton de Gruyter, Berlin/New York (1995)
14. Kathol, A.:Linear syntax. Oxford University Press, Oxford (2000)
15. Nasr, A., Rambow, O.: Supertagging and full parsing. In: Proceedings of the 7th International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+7), pages 56–63, Vancouver, British Columbia, Canada (2004)
16. Neumann, G., Braun, C., Piskorski, J.: A divide-and-conquer strategy for shallow parsing of German free texts. In: Proceedings of the sixth conference on Applied natural language processing (ANLC '00), pp. 239–246. Seatle, WA (2000)

17. Nussbaumer, M.: Rhetorisch-stilistische Eigenschaften der Sprache des Rechtswesens. In: Fix, U., Gardt, A., Knape, J. (eds.) Rhetorik und Stilistik / Rhetoric and Stylistics, Handbooks of Linguistics and Communication Science, vol. 31/2, pp. 2132–2150. Mouton de Gruyter, Boston/New York (2009)
18. Schiller, A., Teufel, C., Stöckert, C., Thielen, C.: Guidelines für das Tagging deutscher Textcorpora mit STTS (kleines und grosses Tagset). Technical report, Universität Stuttgart/Universität Tübingen (1999)
19. Schmid, H.: Improvements in Part-of-Speech Tagging with an Application to German. In: Proceedings of the ACL SIGDAT-Workshop, Dublin (1995)
20. Schneider, G., Volk, M.: Adding Manual Constraints and Lexical Look-Up to a Brill-Tagger for German. In: Proceedings of the ESSLLI-98 Workshop on Recent Advances in Corpus Annotation, Saarbrücken (1998)
21. Volk, M., Schneider, G.: Comparing a Statistical and a Rule-Based Tagger for German. In: Proceeding of the 4th Conference on Natural Language Processing (KONVENS98), pp. 125–137. Peter Lang, Frankfurt a. M., Berlin, Bern, New York, Paris, Wien (1998)
22. Voutilainen, A.: NPtool, a Detector of English Noun Phrases. In: Proceeding of Workshop on Very Large Corpora: Academic and Industrial Perspectives, pp. 48–57. Ohio State University, Columbus (1993)
23. Voutilainen, A.: A Syntax-Based Part-of-Speech Analyser. In: Proceedings of the Seventh Conference on European Chapter of the Association for Computational Linguistics, EACL '95, pp. 157–164. Morgan Kaufmann, San Francisco (1995)