# THE INFLUENCE OF TAGGING ON THE RESULTS OF PARTIAL PARSING IN GERMAN CORPORA

Oliver Wauschkuhn
University of Stuttgart, Institut für Informatik
Breitwiesenstr. 20–22
D-70565 Stuttgart, Germany
e-mail: wauschkuhn@informatik.uni-stuttgart.de

*Published in the Proceedings of the*
*Fourth International Workshop on Parsing Technologies*
*Prague/Karlovy Vary, Czech Republic, September 1995, pp. 260–270.*

## 1 Introduction

In the last two years good progress has been made in work on part-of-speech tagging for German, while for a long time it was focussed on the English language. Now, that German corpora can be tagged with quite a high accuracy (cf. [Schmid 95]), we have to think about for which applications tagging can be used, and how tagged corpora are to be used and processed.

One possibility is the use of a tagger as preprocessor for a partial syntactic analysis of textual corpora. Parsing a sentence results in many cases in a (more or less) great number of ambiguities, which are to be resolved in subsequent analysis steps. To reduce the number of result trees in an earlier stage a tagger could be used to disambiguate the morphological and morphosyntactic part-of-speech annotations of the input for the parser.

This paper describes an examination of the influence of tagging on subsequent partial parsing in German corpora with respect to the number of analysis trees. The underlying question is, to what extent tagging can serve as a preprocessor for a partial syntactic analysis to reduce the number of ambiguities in the parse results. For this, two small corpora have been partially syntactically analyzed, each in a tagged and an untagged form, and the results between these forms are compared with respect to their number of trees.

The term *tagged* corpus(-form)/sentence means in this context, that (most of) its part-of-speech tags are unambiguous. Words of a few classes constitute an exception being annotated with one additional tag if a *structural* analysis is needed to disambiguate them (for more details, see chapter 2). A corpus(-form)/sentence, on the other hand, is called *untagged*, if its words are annotated ambiguously with all tags that result from a morphological analysis without considering their context. This is equal to the input of a tagger.

The next chapter makes some more detailed remarks on the corpus material used for this work and on the part-of-speech annotation. In the third chapter the parsing method for partially analyzing the sentences is explained. In chapter 4 the quantitative results of the different analyses are presented and compared. The last chapter summarizes these results and draws a conclusion with respect to the influence of tagging on the parsing results.

## 2 The Corpus Material

The corpus material used for this evaluation is a small part of the German "Stuttgarter Zeitungs–Corpus", henceforth called StZ-Corpus. The whole corpus consists of the issues of two years of the German newspaper "Stuttgarter Zeitung" and contains ∼36 million tokens, building ∼1.8 million sentences. It has been tagged with the Xerox Part-of-Speech Tagger (cf. [Cutting et al. 92]) adapted to the German language (cf. [Schmid 93]).

One text for this work, consisting of 1097 sentences, is a hand-tagged part of the StZ-Corpus, which served as training- and test-corpus for the adaption of the tagger to German texts. Consequently, its tags are (nearly) 100% correct.

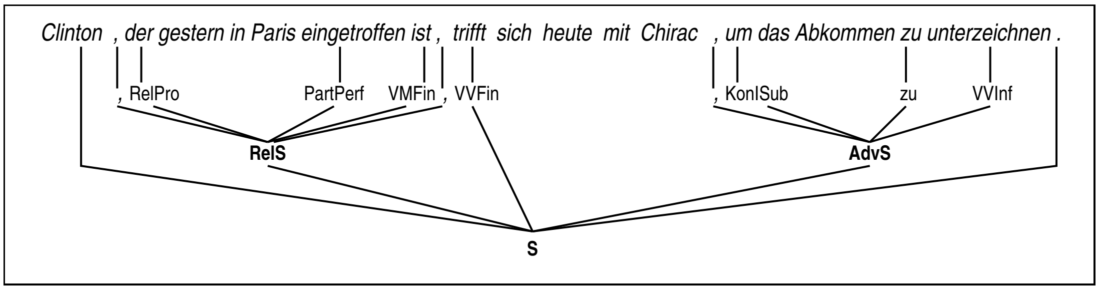


**Figure 1:** Example for the tree structure resulting from the clause-analysis step.

The other text with 3776 sentences is taken from the statistically tagged part of the StZ-Corpus. The tag error rate of the German version of the Xerox tagger can be taken from [Schmid 95] and [SchmidKempe 95] as between 3.5 and 4%. With respect to the whole corpus these values should be seen merely as a lower bound for the error rate, because the text the tagger was tested on was taken from the same corpus part as the training text, and has therefore a similar style of sentence constructions.

The tagset the StZ-Corpus is tagged with consists of 71 tags representing syntactic categories, which are grouped into the following 12 main categories: noun, adjective, cardinal number, verb, determiner, pronoun, adverb, conjunction, adposition (preposition, postposition, etc.), structural word, interjection, punctuation mark. For detailed information about this tagset, see [Schiller 94], and an overview is given in [SchillerThielen 95]. In addition to these syntactic tags, the hand tagged text is also annotated with morphosyntactic feature information, like case, gender, number, etc.

Auxiliaries and modal verbs are *never* annotated as main verbs in the tagged StZ-Corpus, even if they have such a function in the sentence. Generally, a bigram tagger, which considers only the left context of the word to be annotated, cannot decide between these two purposes of auxiliaries and modal verbs: in a German main clause construction, an auxiliary/modal verb and the corresponding main verb often do not appear adjacently—i.e. they build discontinuous constituents—, and the main verb stands in the right context of the corresponding auxiliary/modal verb.

As the grammars for the partial parsing process make use of the distinction between an auxiliary/modal use of an auxiliary/modal verb and the main verb use, all those verbs in the tagged corpora had to be annotated with an additional main verb tag before the partial syntactic analysis.

## 3 The Partial Parsing Process

### 3.1 Method

For the domain of a partial syntactic analysis of German textual corpora we developed a method that divides the parse process into two steps. The task of the first step is to detect all single clauses of the possibly complex input sentence, i.e. main clauses, subordinate clauses and infinitive constructions. Their hierarchical order (coordination, subordination) is here of little interest. This analysis step is henceforth also called *clause-analysis*. An example for the result structure of this step is shown in Figure 1 for the following sentence:[1]

> *Clinton, der gestern in Paris eingetroffen ist, trifft sich heute mit Chirac, um das Abkommen zu unterzeichnen.*
> (*Clinton, who arrived in Paris yesterday, meets Chirac today to sign the agreement.*)

It consists of a main clause (S) (*"Clinton trifft sich heute mit Chirac."*), a relative clause (RelS) (*", der gestern in Paris eingetroffen ist,"*) and an adverbial clause (AdvS) (*", um das Abkommen zu unterzeichnen"*).

[1]The translation is given in parentheses.

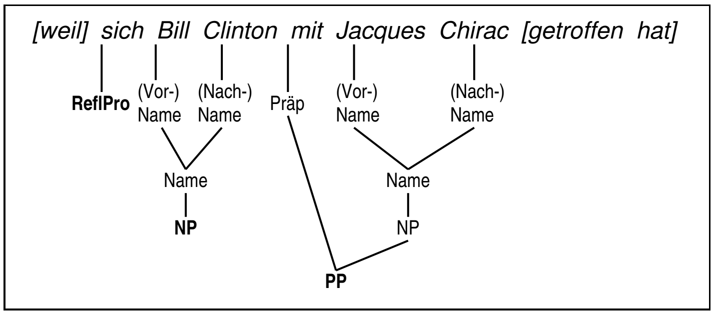


**Figure 2:** Example for the tree structure resulting from the NP-analysis step.

In the second step, the minimal NPs and PPs[2] within each single clause of the complete sentence are to be detected without making any decision about their hierarchical order (coordination, subordination). I.e., inside a single clause all NPs and PPs are on the same hierarchical level. Henceforth, this step is also referred to as *NP-detection* or *NP-analysis*. An example of the result of this parse step is given in Figure 2; the adverbial clause

> *[weil] sich Bill Clinton mit Jacques Chirac [getroffen hat]*
> (*[because] Bill Clinton [has met] Jacques Chirac*)

consists of a reflexive pronoun (ReflPro) (*"sich"*), a noun phrase (NP) (*"Bill Clinton"*) and a prepositional phrase (PP) (*"mit Jacques Chirac"*).

To parse a sentence partially, these two steps are applied successively: First, the clause-analysis is done and then the NP-detection is applied to every single clause of the result of the first step. The second step is not carried out for sentences that do not yield a result in the clause-analysis step. The combination of the outcome of all these parses for a sentence builds the final result of the partial analysis.

The division of the parsing process into the clause- and the NP-analysis is especially adequate for German sentence constructions. Complex sentences are well structured into clauses by means of punctuation marks (especially commas) and structural words (especially different types of conjuctions), and simple minimal NPs do not extend clause limits. In addition, the independence between these parse steps makes the method more robust. A detailed description of this stepwise parsing strategy is given in [Wauschkuhn 94].

### 3.2 Realization

The parse process is realized by means of a chart parser (CHAPLIN)[3], which applies sets of syntactic rules (grammars) on annotated input sequences. The grammars are of a context-free type and their rules can additionally define morphosyntactic feature restrictions between the right-hand side categories. The input for an analysis is an ordered set of categorially and morphosyntactically annotated wordforms.

The development of the grammar for the first analysis step is in an advanced stadium, while the grammar of the NP-detection consists mainly of a set of "basic rules" and has to be further developed and refined.

The clause-analysis grammar consists of 229 phrase structure rules, the one for the NP-detection of 180.

## 4 Results from the Parsing Process

### 4.1 Description of the Experiment

Both texts described in chapter 2—the hand-tagged one, consisting of 1097 sentences, henceforth also referred to as *Corpus 1*, and the statistically tagged one with 3776 sentences, henceforth

[2] noun phrases and prepositional phrases

[3] The parser is implemented in COMMON LISP and described in detail in [BurkertLöthe 95].

also called *Corpus 2*—have been analyzed in the following way:

Each corpus was partially parsed twice along the two-step method, once in the tagged form[4] and once in the untagged form. The single words of the untagged corpus versions were analyzed with two German morphology systems (GERMMORPHAN, implemented by E. Lehmann, University of Stuttgart, and MORPH, implemented by G. Hanrieder, FORWISS[5], Erlangen) and then were annotated with the resulting categories and morphosyntactic features. So, the words of the untagged version of Corpus 2 bore a bit more information than in the tagged form, namely the morphosyntactic features.

Since (partial) syntactically (hand-)annotated *German* corpora—like the Penn Treebank for the English language—are not yet (freely) available, it was impossible to automatically evaluate parsing results for quality. Therefore, a quantitative evaluation was made considering the number of parse trees resulting from the syntactic analysis of the input sequences. The results for each corpus form (tagged and untagged) of both corpora are represented in detail within Tables 1a–4b in the appendix. Each table holds for the different numbers of resulting parse trees (column 1), the corresponding number of input sequences that yields this tree number (column 2), and its percentage with respect to the complete number of inputs (column 3).

In addition, it has been examined for the clause-analysis step, how many sentences yield the same quantitative non-ambiguous parse result (i.e. 0 and 1 tree, respectively) for *both* input forms, tagged and untagged. This amount compared to the number of sentences that yield the same result for their untagged form represents a measure for the disambiguating quality of the tagger for subsequent partial parsing: this proportion indicates the relative amount (percentage) of sentences that were annotated with such parts-of-speech by the tagger that they yield the same non-ambiguous parse result (0 or 1 tree) as in their untagged form. So, the same percentage for selecting the "correct" tags with respect to a subsequent syntactic analysis can be expected for the other cases where the parse process yields more than one tree. This measure of quality can be put into numbers through the quotient of the amount of sentences having the same quantitative analysis result for both corpus versions and the amount of sentences yielding that result for the untagged version. These values are represented in Table 6 in the appendix.

The input type for the clause-analysis is a sentence and the input for the second step is a sequence of words from a single clause that is limited to the left and to the right by elements of type "verb" or "clause boundary". Thus, in most cases for successfully clause-analyzed sentences, more than one NP-analysis is to be made.

Both analysis steps are evaluated seperately for this work. If a clause-analysis results in more than one parse tree, the second step is evaluated only for the simple clauses of *one* of the results of step 1. Thus each sentence part is NP-analyzed only one time and is not considered more than once in the results.

To guarantee that the parser won't get stuck while analyzing a word sequence and building the corresponding parse forest[6] for counting the number of result trees, two interruption criteria are defined: *a)* a time limit is defined at which the computation is interrupted, and *b)* for the untagged corpus versions input sequences of more than 50 elements for the first step and of more than 11 elements for the second step are ignored, i.e. not analyzed. Those cases are listed in the tables in the first row "interruption".

### 4.2 Results of each Analysis

Now that all conditions of the parse experiment have been explained, the results of the analyses for the different corpus types are to be presented and compared.

First, the results of the *clause-analyses* of the four corpus types (Corpus 1 and 2, each in the tagged and the untagged version) will be described and discussed and after that the results of the *NP-detection*. All quantitative results are summarized in Tables 1a–4b. The comparisons are made between tagged and untagged version of each corpus.

[4]As mentioned, the auxiliaries and modal verbs are co-tagged as main verbs in the tagged corpus versions.

[5]Bayerisches Forschungszentrum für Wissensbasierte Systeme

[6]The time- and space-critical phases of an analysis of "hard" input sequences is generally *not* the parse process itself, i.e. the filling of the chart, but the construction of the parse forest from the chart.

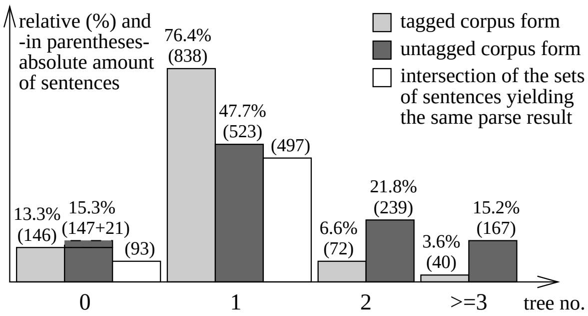


**Chart 1:** Amounts of sentences resulting in 0, 1, 2, and ≥3 parse trees in the clause-analysis step for Corpus 1.

The essential analysis results relevant to the following discussions are represented in several bar charts (Chart 1–5). In addition to the amounts of resulting parse trees, they show for the first analysis step the amounts of sentences that yield the same quantitative result for the tagged and the untagged corpus version. In the charts for the NP-analysis the absolute result values (tree numbers) are not contained, because the number of analyzed input sequences differs between the tagged and the untagged corpus versions. Thus, only the relative comparison is of interest.

### Clause-Analysis

*Corpus 1 — The Hand-Tagged Text*

All quantitative results of the clause-analysis of the hand-tagged corpus are given in detail in Tables 1a and 1b for the tagged and the untagged version respectively; Chart 1 gives an overview. Both corpus forms have nearly the same amount of sentences that are not covered by the grammar for this parse step, namely *13.3%* and *13.4%*. In addidion, for the untagged version the parse process was interrupted or not carried out for *1.9%* of the input[7], leading to a rate of *15.3%* for not successfully parsed sentences.

The intersection of the set of tagged sentences that lead to no parse result and the corresponding set of untagged sentences contains only *63.2%* of the untagged set of unparsable sentences (see also Table 5 in the appendix).

Actually, one would expect a much higher correspondence between these sets of input, especially, because they have nearly the same number of elements (146 and 147 respectively): If a parser finds a parse tree for an unambiguously annotated (i.e. tagged) sentence, it should also find one or more results for it in the ambiguously annotated (untagged) version, because the disambiguated tags should be contained in the non-disambiguated tags. And if the parser doesn't find a parse tree for an ambiguously annotated sentence, how should a result be found for the disambiguated form?

There are two reasons for this difference between the sets of unparsable sentences: First, the hand-tagging of the corpus was not done on the basis of the results of the morphology systems used to analyze the untagged version, and second, even the morphology systems make a few mistakes, eg. for some unknown verbforms.

Considering the successful analysis results presumes a great advantage of tagging a corpus before parsing it with respect to the number of results: *76.4%* of the sentences in the hand-tagged corpus form led to an unambiguous analysis result while the corresponding rate for the untagged version is only *47.7%*. But in this case, it must be taken into account that the tagged corpus form has an error rate of nearly 0% due to manual annotation.

In contrast to the unsuccessful parses, the comparison of the sets of sentences (tagged vs. untagged) yielding one analysis result confirms the expectation: *95.0%* of the untagged sentences led also in their tagged form to one unambiguous result (cf. also Table 5).

[7]This concerns sentences with more than 50 elements or needing a computation time of more than 5 minutes.

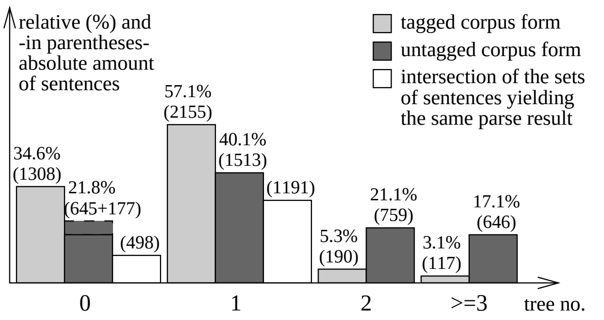


**Chart 2:** Amounts of sentences resulting in 0, 1, 2, and ≥3 parse trees in the clause-analysis step for Corpus 2.

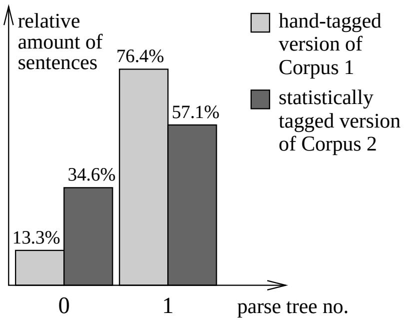


**Chart 3:** Comparison of quantitative analysis results between manually and statistically tagged sentences.

More than one parse tree was found for *10.2%* of the hand-tagged sentences (*6.6%* for 2 trees + *3.6%* for 3 or more) and for a relatively great part of the untagged sentences, namely *37.0%* (*21.8%* + *15.2%*).

*Corpus 2 — The Statistically Tagged Text*

Tables 2a and 2b hold detailled quantitative results for the clause-analysis of Corpus 2; see Chart 2 for an overview. *34.6%* of the statistically tagged sentences were not covered by the grammar for the first analysis step while this holds for only *17.1%* of the untagged sentences, plus *4.7%* for interrupted analyses. This great difference between the results of tagged and untagged input resulting in 0 parse trees didn't arise for the manually annotated Corpus 1 and is caused for a certain part by wrong tags produced by the statistical tagger (see below).

Comparing the sets of tagged and untagged sentences yielding an unsuccessful parse result shows that *77.2%* of the untagged sentences are also contained in the unparsable tagged sentence set (see Table 5).

Considering the unambiguous successful results (i.e., exactly one parse tree was found) also emphasizes the disambiguating quality of tagging with regard to the syntactic analysis, however, not to such an extent as in the hand-tagged case: *57.1%* of non-ambiguous results for the statistically tagged version versus *40.1%* for the untagged. Similary to the unsuccessful parses, *78.7%* of the unambiguously analyzed untagged sentences yielded also in the tagged form one parse tree.

The cases for 2 or more parse trees resemble to the results for Corpus 1.

*Hand-Tagged vs. Statistically Tagged Input*

A comparison between the clause-analysis results of the manually and the statistically tagged corpus versions (see Chart 3 for an overview and Tables 1a and 2a respectively for detailed results) shows that in the latter version *21.3%* more sentences resulted in 0 parse trees than in the former, while on the other hand the percentage of successfully unambiguously parsed sentences is less by a similar amount (*19.3%*).

This shift from successful unambiguous parse results to unsuccessful analyses between these corpus versions supports the statement from above that a certain part of the unsuccessfully parsed sentences is caused by wrong tags relevant to this analysis step. A closer look at the first 30 of all 1309 non-parsable sentences of the (statistically) tagged form of Corpus 2 led to the following result: for 21 sentences (*70.0%*) the failure was due to one wrong tag, 7 sentences (*23.3%*) were not covered by the grammar and 2 sentences (*6.7%*) had both shortcomings.

However, even if the rate is worse, statistical tagging helps to reduce the number of ambiguities in the results of subsequent partial parsing.

*Some Remarks on the Imbalance in the Results Between Tagged and Untagged Corpus Forms*

To explain the imbalance in the results between tagged and untagged input, one must have a look at the categories that are relevant to the first analysis step (i.e. for the detection of single clauses in complex sentences). The main group of these are verbs and function words. Many German function words have more than one possible syntactic category depending on their use in clauses/sentence constructions. Thus, the morphology produces more than one tag for them. Some examples:

*"zu"* → infinitive introductory structural word, preposition, separable verb prefix
*"und"* → phrase coordinating conjunction, clause coordinating conjunction
*"wie"* → preposition, comparative conjunction, clause subordin. conjunction, interrogative adverb

A tagger reduces these part-of-speech ambiguities by considering the near context of the words which in turn reduces the number of possible syntactic structures. On the other hand, the tagger is not able to decide generally about parts-of-speech that are based on long-distance dependencies, which can cause wrong parse results.

**NP-Analysis**

As already mentioned above, the development of the grammar for the second analysis step is less advanced than for the first step, which can also be derived from the success rates as well as from the amount of the following quantitative parse results. Therefore, these shall be seen merely with regard to a *comparison* between tagged and untagged input.

*Corpus 1 — The Hand-Tagged Text*

Quantitative results for the NP-analysis of the hand-tagged corpus are given in Tables 3a for the tagged and in 3b for the untagged form; Chart 4 gives an overview. A striking feature of the tagged version is the high rate of *41.4%* of not successfully parsed input sequences, while for the untagged corpus form this rate is at *22.4%*, plus *8.2%* for interrupted parses.

This great amount of unparsable hand-tagged input is in parts caused by the bad tuning of the NP-grammar to the tagset the corpus is annotated with. The grammar is tuned to the morphology results used in the untagged input, which have for example a finer distinction in the noun categories than the tagset of the tagger. While in the latter there exists only the category noun, the morphological analysis distinguishes between uncountable nouns (eg. *rice, butter, water*), nouns representing a quantity unit (eg. *litre, million*), and other "normal" nouns. Thus, in the grammar, a number expression followed by a noun can only combine to a NP, if the noun is an uncountable noun, a quantity unit or in plural number; but that combination is not possible with a singular "normal" noun.

For the case of successfully unambiguously analyzed input sequences quite a high rate of *52.0%* could be found for the hand-tagged corpus form while the corresponding rate for untagged input is much less, namely *18.6%*. On the other hand, the great amount of *50.8%* of untagged input sequences yielded an ambiguous parse result; *39.6%* produced 3 or more trees and even *13.5%* 10 or more. This high ambiguity rate for untagged input is above all due to the various functions German determiner words can have (*"der"*, *"die"*, *"das"* → determiner, relative pronoun, demonstrative pronoun; *"eine"*, *"einen"* → determiner, indefinite pronoun), and the fact that nouns—especially not so often used ones—are frequently classified also as names by the morphology. Such ambiguities produce a great variety of different potential chains of NPs for the input sequences.

Especially for these cases tagging is a useful tool to highly reduce the number of parse trees, because dependencies between categories relevant to the NP-analysis extend mainly over a short distance. However, these results still have to be examined with respect to quality, i.e. correctness of the annotations.

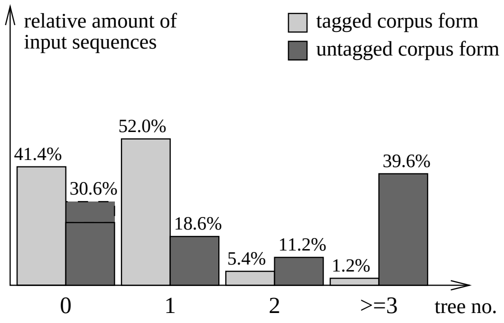


**Chart 4:** Amounts of sentences resulting in 0, 1, 2, and ≥3 parse trees in the NP-analysis step for Corpus 1.

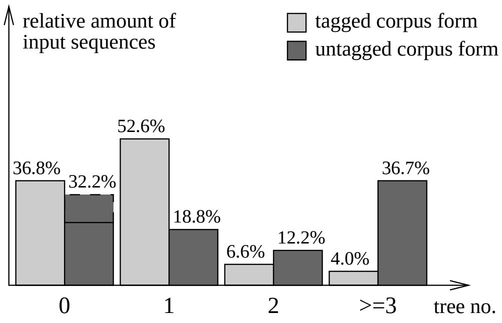


**Chart 5:** Amounts of sentences resulting in 0, 1, 2, and ≥3 parse trees in the NP-analysis step for Corpus 2.

*Corpus 2 — The Statistically Tagged Text*

Considering the results of the NP-analysis of the statistically tagged corpus in its tagged and untagged versions (Tables 4a and 4b respectively and Chart 5), a great similarity to the quantitative parse results of Corpus 1 can be noticed. The only difference that is worth mentioning is, that the amount of unsuccessfully parsed sentences (0 trees) is *4.6%* smaller for the statistically tagged corpus form than for the hand-tagged one (Corpus 1), and that this part has been shifted to the results of 2 or more analysis trees. This shift can be explained by the lack of morphosyntactic annotations in the statistically tagged corpus.[8] Due to this lack, the great number of feature restrictions in the NP-grammar is not considered in the parse process, so that more phrase structure rules are applied than actually match the syntactic constructions. This results in a greater number of syntax trees having a certain error rate.

Since all other results are similar to those of the NP-analyses of Corpus 1, nothing new can be said upon the comparison between the parse results of the tagged and the untagged corpus version.

## 5 Summary and Conclusion

A very conspicuous property among the quantitative results is that for each *tagged* corpus version in both analysis steps the amount of unambiguously parsed input sequences lies over *50%*—for the clause-analysis of hand-tagged input even over *75%*—, while on the other hand, the correponding rate for *untagged* input is about *40%* and *47%* for the clause-analysis and below *20%* for the NP-analysis.

Considering the amounts of unsuccessful parse results (i.e. 0 trees) shows in most cases a much higher rate for tagged than for untagged input.

A conclusion from these facts and numbers is that tagging helps to reduce the number of parse trees but at the cost of increasing the rate of unsuccessful parses.

For lack of manually syntactically annotated German corpora, parse results couldn't be proved for correctness on a large scale automatically. Therefore no details can be given about the *quality* of the disambiguating character of the tagger for a subsequent syntactic analysis, which in turn depends on the correctness of the part-of-speech annotations.

However, the syntactic results were examined "indirectly" at least for the clause-analysis step by comparing the sets of sentences that result in the same number of parse trees (for 0 and 1) between tagged and untagged input. For the case of Corpus 2 *78,7%* of sentences that yielded in the clause-analysis exactly one parse tree for their tagged form resulted in the same tree number for their statistically tagged version. For this amount of sentences the tagger has choosen the "correct"[9] parts-of-speech relevant to the first analysis step, and a similar

[8]Recall that the untagged version of Corpus 2 is in contrast morphosyntactically annotated for the parse process.

[9]"correct" only with respect to yield a parse result.

percentage of correctness can be expected for the other statistically annotated sentences.

Regarding the relatively high rate of unsuccessful parse results for tagged input, two reasons can be given: One problem is the error rate of the tagger of at least 3.5–4%, which has been confimed by the small examination of the tags of the non-clause-analysable sentences of the statistically tagged corpus, where in *70%* of the sentences a wrong tag caused the non-parsability. The second problem effecting the NP-analysis is the lacking adaption of the NP-grammar to the tags of the tagger.

In both cases improvements can be made: In the tagged corpus, frequently repeated tag errors must be recognized, and either the tagger is to be improved with respect to these cases or those tags are to be co-tagged with all possible parts-of-speech for the corresponding word. For the NP-grammar, refinements and extensions in the rule set are to be made.

To use a tagger as preprocessor for syntactical analyses, only such parts-of-speech should be disambiguated with it that can be decided on the basis of the considered context. For a bigram- or trigram-tagger this is the case for short-distance dependencies, and long-distance phenomena like combinations of an auxiliary with the corresponding participle are to be resolved by the parser. Therefore, a tagger is especially useful to disambiguate such words that are relevant to the partial NP-analysis: their parts-of-speech depend in many cases on adjacent words/categories, and furthermore, as mentioned above (4.2: NP-Analysis), these words can produce a great amount of syntactic ambiguities.

To conclude I want to give a proposal on how to use a tagger as a preprocessor for parsing a textual corpus:

1. For the case of a successful syntactic analysis of a tagged sentence, the analysis result should be accepted as a disambiguated parse forest.

2. Tagging errors increase the number of unsuccessful parses. In such a case, i.e., if the syntactic analysis of a tagged sentence results in 0 trees, this sequence should be reparsed in its untagged form.

This method makes use of a tagger as a tool for disambiguating syntactic results only in the advantageous cases. Otherwise the tagger results are ignored.

## References

[BurkertLöthe 95] Gerrit Burkert and Mathis Löthe. ChaPLin – Ein Chart Parser für Linguistische Experimente. Technical Report, Institut für Informatik, University of Stuttgart, 1995, forthcoming.

[Cutting et al. 92] Doug Cutting and Julian Kupiec and Jan Pedersen and Penelope Sibun. A Practical Part-of-Speech Tagger. In *Proceedings of the third Conference on Applied Natural Language Processing*, Trento, Italy, April 1992.

[Schiller 94] Anne Schiller. Guidelines für das Tagging deutscher Textcorpora (Kleines und erweitertes Tagset). Technical Report, Institut für maschinelle Sprachverarbeitung (IMS), University of Stuttgart, 1994.

[SchillerThielen 95] Anne Schiller and Christine Thielen. Ein kleines und erweitertes Tagset fürs Deutsche. In *Tagungsberichte des Arbeitstreffens "Lexikon + Text", Schloß Hohentübingen, 1994.* Niemeyer, Tübingen, 1995.

[Schmid 93] Helmut Schmid. Tagging German with the Xerox-Tagger. Technical Report, Institut für maschinelle Sprachverarbeitung (IMS), University of Stuttgart, 1993.

[Schmid 95] Helmut Schmid. Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of EACL SIGDAT-Workshop, Dublin, Ireland, 1995.*

[SchmidKempe 95] Helmut Schmid and André Kempe. Tagging von Corpora mit HMM, Entscheidungsbäumen und Neuronalen Netzen. In *Tagungsberichte des Arbeitstreffens "Lexikon + Text", Schloß Hohentübingen, 1994.* Niemeyer, Tübingen, 1995.

[Wauschkuhn 94] Oliver Wauschkuhn. Mehrstufiges Parsing zur syntaktischen Analyse von Textcorpora. Technical Report, Institut für Informatik, University of Stuttgart, 1994.

## Appendix: Tables

| *no. of parse trees* | *no. of sentences* | *relative amount of sentences* |
|---|---|---|
| interrupt. | 1 | 0.0% |
| 0 | 146 | 13.3% |
| 1 | 838 | 76.4% |
| 2 | 72 | 6.6% |
| 3 | 11 | 1.0% |
| 4 | 11 | 1.0% |
| 5 | 1 | 0.0% |
| 6–9 | 10 | 0.9% |
| 10–19 | 6 | 0.5% |
| 20–49 | 1 | 0.0% |
| ≥50 | 0 | 0.0% |
| *sum* | *1097* | |

**Table 1a:** Hand-tagged input.

| *no. of parse trees* | *no. of sentences* | *relative amount of sentences* |
|---|---|---|
| interrupt. | 21 | 1.9% |
| 0 | 147 | 13.4% |
| 1 | 523 | 47.7% |
| 2 | 239 | 21.8% |
| 3 | 41 | 3.7% |
| 4 | 46 | 4.2% |
| 5 | 3 | 0.3% |
| 6–9 | 42 | 3.8% |
| 10–19 | 27 | 2.5% |
| 20–49 | 8 | 0.7% |
| ≥50 | 0 | 0.0% |
| *sum* | *1097* | |

**Table 1b:** Untagged input.

**Absolute and relative amounts of sentences in relation to the number of parse trees they result in for the clause-analysis of Corpus 1.**

| *no. of parse trees* | *no. of sentences* | *relative amount of sentences* |
|---|---|---|
| interrupt. | 6 | 0.2% |
| 0 | 1308 | 34.6% |
| 1 | 2155 | 57.1% |
| 2 | 190 | 5.3% |
| 3 | 23 | 0.6% |
| 4 | 34 | 0.9% |
| 5 | 9 | 0.2% |
| 6–9 | 27 | 0.7% |
| 10–19 | 16 | 0.4% |
| 20–49 | 8 | 0.2% |
| ≥50 | 0 | 0.0% |
| *sum* | *3776* | |

**Table 2a:** Statistically tagged input.

| *no. of parse trees* | *no. of sentences* | *relative amount of sentences* |
|---|---|---|
| interrupt. | 177 | 4.7% |
| 0 | 645 | 17.1% |
| 1 | 1513 | 40.1% |
| 2 | 795 | 21.1% |
| 3 | 140 | 3.7% |
| 4 | 222 | 5.9% |
| 5 | 31 | 0.8% |
| 6–9 | 125 | 3.3% |
| 10–19 | 76 | 2.0% |
| 20–49 | 45 | 1.2% |
| ≥50 | 7 | 0.2% |
| *sum* | *3776* | |

**Table 2b:** Untagged input.

**Absolute and relative amounts of sentences in relation to the number of parse trees they result in for the clause-analysis of Corpus 2.**

| *no. of parse trees* | *no. of input sequ.* | *relative amount of input sequ.* |
|---|---|---|
| interrupt. | 0 | 0.0% |
| 0 | 1244 | 41.4% |
| 1 | 1562 | 52.0% |
| 2 | 162 | 5.4% |
| 3 | 8 | 0.3% |
| 4 | 17 | 0.6% |
| 5 | 1 | 0.0% |
| 6–9 | 6 | 0.2% |
| 10–19 | 3 | 0.1% |
| 20–49 | 1 | 0.0% |
| 50–99 | 0 | 0.0% |
| ≥100 | 0 | 0.0% |
| *sum* | *3004* | |

**Table 3a:** Hand-tagged input.

| *no. of parse trees* | *no. of input sequ.* | *relative amount of input sequ.* |
|---|---|---|
| interrupt. | 230 | 8.2% |
| 0 | 629 | 22.4% |
| 1 | 524 | 18.6% |
| 2 | 315 | 11.2% |
| 3 | 388 | 13.8% |
| 4 | 120 | 4.3% |
| 5 | 27 | 1.0% |
| 6–9 | 199 | 7.1% |
| 10–19 | 198 | 7.0% |
| 20–49 | 106 | 3.8% |
| 50–99 | 50 | 1.8% |
| ≥100 | 27 | 1.0% |
| *sum* | *2813* | |

**Table 3b:** Untagged input.

**Absolute and relative amounts of input sequences in relation to the number of parse trees they result in for the NP-analysis of Corpus 1.**

| *no. of parse trees* | *no. of input sequ.* | *relative amount of input sequ.* |
|---|---|---|
| interrupt. | 2 | 0.0% |
| 0 | 2804 | 36.8% |
| 1 | 4011 | 52.6% |
| 2 | 504 | 6.6% |
| 3 | 84 | 1.1% |
| 4 | 116 | 1.5% |
| 5 | 30 | 0.4% |
| 6–9 | 55 | 0.7% |
| 10–19 | 17 | 0.2% |
| 20–49 | 3 | 0.0% |
| 50–99 | 0 | 0.0% |
| ≥100 | 0 | 0.0% |
| *sum* | *7626* | |

**Table 4a:** Statistically tagged input.

| *no. of parse trees* | *no. of input sequ.* | *relative amount of input sequ.* |
|---|---|---|
| interrupt. | 802 | 8.5% |
| 0 | 2220 | 23.7% |
| 1 | 1765 | 18.8% |
| 2 | 1148 | 12.2% |
| 3 | 1170 | 12.5% |
| 4 | 356 | 3.8% |
| 5 | 82 | 0.9% |
| 6–9 | 684 | 7.3% |
| 10–19 | 528 | 5.6% |
| 20–49 | 337 | 3.6% |
| 50–99 | 162 | 1.7% |
| ≥100 | 129 | 1.4% |
| *sum* | *9383* | |

**Table 4b:** Untagged input.

**Absolute and relative amounts of input sequences in relation to the number of parse trees they result in for the NP-analysis of Corpus 2.**

| | *Corpus 1* | | *Corpus 2* | |
|---|---|---|---|---|
| | *0 trees* | *1 tree* | *0 trees* | *1 tree* |
| compared sentence numbers | 93 of 147 | 497 of 523 | 498 of 645 | 1191 of 1513 |
| relative amount by division | 63.2% | 95.0% | 77.2% | 78.7% |

**Table 5:** Amounts of sentences that yield the same number of parse trees (in the clause-analysis step) for both input forms, tagged and untagged (for 0 and 1 tree), compared to the amount of sentences that result in this tree number for the untagged form.