

Guiding a Constraint Dependency Parser with Supertags

Kilian Foth, Tomas By, and Wolfgang Menzel

Department für Informatik, Universität Hamburg, Germany
foth|by|menzel@informatik.uni-hamburg.de

Abstract

We investigate the utility of supertag information for guiding an existing dependency parser of German. Using weighted constraints to integrate the additionally available information, the decision process of the parser is influenced by changing its preferences, without excluding alternative structural interpretations from being considered. The paper reports on a series of experiments using varying models of supertags that significantly increase the parsing accuracy. In addition, an upper bound on the accuracy that can be achieved with perfect supertags is estimated.

1 Introduction

Supertagging is based on the combination of two powerful and influential ideas of natural language processing: On the one hand, parsing is (at least partially) reduced to a decision on the optimal sequence of categories, a problem for which efficient and easily trainable procedures exist. On the other hand, supertagging exploits complex categories, i.e. tree fragments which much better reflect the mutual compatibility between neighbouring lexical items than say part-of-speech tags.

Bangalore and Joshi (1999) derived the notion of supertag within the framework of Lexicalized Tree-Adjoining Grammars (LTAG) (Schabes and Joshi, 1991). They considered supertagging a process of almost parsing, since all that needs to be done after having a sufficiently reliable sequence of supertags available is to decide on their combination into a spanning tree for the complete sentence. Thus the approach lends itself easily to pre-processing sentences or filtering parsing results with the goal of guiding the parser or reducing its output ambiguity.

Nasr and Rambow (2004) estimated that perfect supertag information already provides for a parsing accuracy of 98% if a correct supertag assignment were available. Unfortunately, perfectly reliable supertag information cannot be expected; usually this uncertainty is compensated by running the tagger in multi-tagging mode, expecting that the reliability can be increased by not forcing the tagger to take unreliable decisions but instead offering a set of alternatives from which a subsequent processing component can choose.

A grammar formalism which seems particularly well suited to decompose structural descriptions into lexicalized tree fragments is dependency grammar. It allows us to define supertags on different levels of granularity (White, 2000; Wang and Harper, 2002), thus facilitating a fine grained analysis of how the different aspects of supertag information influence the parsing behaviour. In the following we will use this characteristic to study in more detail the utility of different kinds of supertag information for guiding the parsing process.

Usually supertags are combined with a parser in a filtering mode, i.e. parsing hypotheses which are not compatible with the supertag predictions are simply discarded. Drawing on the ability of Weighted Constraint Dependency Grammar (WCDG) (Schröder et al., 2000) to deal with defensible constraints, here we try another option for making available supertag information: Using a *score* to estimate the general reliability of unique supertag decisions, the information can be combined with evidence derived from other constraints of the grammar in a soft manner. It makes possible to rank parsing hypotheses according to their plausibility and allows the parser to even override potentially wrong supertag decisions.

Starting from a range of possible supertag models, Section 2 explores the reliability with which dependency-based supertags can be determined on

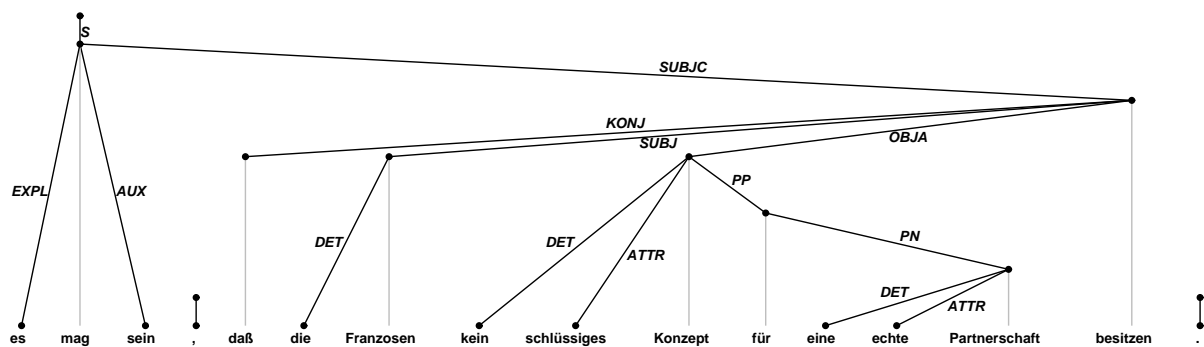


Figure 1: Dependency tree for sentence 19601 of the NEGRA corpus.

different levels of granularity. Then, Section 3 describes how supertags are integrated into the existing parser for German. The complex nature of supertags as we define them makes it possible to separate the different structural predictions made by a single supertag into components and study their contributions independently (c.f. Section 4). We can show that indeed the parser is robust enough to tolerate supertag errors and that even with a fairly low tagger performance it can profit from the additional, though unreliable information.

2 Supertagging German text

In defining the nature of supertags for dependency parsing, a trade-off has to be made between expressiveness and accuracy. A simple definition with very small number of supertags will not be able to capture the full variety of syntactic contexts that actually occur, while an overly expressive definition may lead to a tag set that is so large that it cannot be accurately learnt from the training data. The local context of a word to be encoded in a supertag could include its edge label, the attachment direction, the occurrence of obligatory¹ or of all dependents, whether each predicted dependent occurs to the right or to the left of the word, and the relative order among different dependents. The simplest useful task that could be asked of a supertagger would be to predict the dependency relation that each word enters. In terms of the WCDG formalism, this means associating each word at least with one of the syntactic labels that decorate dependency edges, such as SUBJ or DET; in other words, the supertag set would be identical to the label set. The example sentence

¹The model of German used here considers the objects of verbs, prepositions and conjunctions to be obligatory and most other relations as optional. This corresponds closely to the set of needs roles of (Wang and Harper, 2002).

“Es mag sein, daß die Franzosen kein schlüssiges Konzept für eine echte Partnerschaft besitzen.”

(Perhaps the French do not have a viable concept for a true partnership.)

if analyzed as in Figure 1, would then be described by a supertag sequence beginning with EXPL S AUX . . .

Following (Wang and Harper, 2002), we further classify dependencies into Left (L), Right (R), and No attachments (N), depending on whether a word is attached to its left or right, or not at all. We combine the label with the attachment direction to obtain composite supertags. The sequence of supertags describing the example sentence would then begin with EXPL/R S/N AUX/L . . .

Although this kind of supertag describes the role of each word in a sentence, it still does not specify the entire local context; for instance, it associates the information that a word functions as a subject only with the subject and not with the verb that takes the subject. In other words, it does not predict the relations *under* a given word. Greater expressivity is reached by also encoding the labels of these relations into the supertag. For instance, the word ‘mag’ in the example sentence is modified by an expletive (EXPL) on its left side and by an auxiliary (AUX) and a subject clause (SUBJC) dependency on its right side. To capture this extended local context, these labels must be encoded into the supertag. We add the local context of a word to the end of its supertag, separated with the delimiter +. This yields the expression S/N+AUX, EXPL, SUBJC. If we also want to express that the EXPL precedes the word but the AUX follows it, we can instead add two new fields to the left and to the right of the supertag, which leads to the new supertag EXPL+S/N+AUX, SUBJC.

Table 1 shows the annotation of the example us-

Word	Supertag model J
es	+EXPL/R+
mag	EXPL+S/N+AUX,SUBJC
sein	+AUX/L+
,	+/N+
daß	+KONJ/R+
die	+DET/R+
Franzosen	DET+SUBJ/R+
kein	+DET/R+
schlüssiges	+ATTR/R+
Konzept	ATTR,DET+OBJA/R+PP
für	+PP/L+PN
eine	+DET/R+
echte	+ATTR/R+
Partnerschaft	ATTR,DET+PN/L+
besitzen	KONJ,OBJA,SUBJ+SUBJC/L+
.	+/N+

Table 1: An annotation of the example sentence

ST model	label	direction	dependents	order	#tags	Supertag accuracy	Component accuracy
A	yes	no	none	no	35	84.1%	84.1%
B	yes	yes	none	no	73	78.9%	85.7%
C	yes	no	oblig.	no	914	81.1%	88.5%
D	yes	yes	oblig.	no	1336	76.9%	90.8%
E	yes	no	oblig.	yes	1465	80.6%	91.8%
F	yes	yes	oblig.	yes	2026	76.2%	90.9%
G	yes	no	all	no	6858	71.8%	81.3%
H	yes	yes	all	no	8684	67.9%	85.8%
I	yes	no	all	yes	10762	71.6%	84.3%
J	yes	yes	all	yes	12947	67.6%	84.5%

Table 2: Definition of all supertag models used.

ing the most sophisticated supertag model. Note that the notation +EXPL/R+ explicitly represents the fact that the word labelled EXPL has no dependents of its own, while the simpler EXPL/R made no assertion of this kind. The extended context specification with two + delimiters expresses the complete set of dependents of a word and whether they occur to its left or right. However, it does not distinguish the order of the left or right dependents among each other (we order the labels on either side alphabetically for consistency). Also, duplicate labels among the dependents on either side are not represented. For instance, a verb with two post-modifying prepositions would still list PP only once in its right context. This ensures that the set of possible supertags is finite. The full set of different supertag models we used is given in Table 2. Note that the more complicated models G, H, I and J predict all dependents of each word, while the others predict obligatory dependents only, which should be an easier task.

To obtain and evaluate supertag predictions, we used the NEGRA and TIGER corpora (Brants et al., 1997; Brants et al., 2002), automatically trans-

formed into dependency format with the freely available tool DepSy (Daum et al., 2004). As our test set we used sentences 18,602–19,601 of the NEGRA corpus, for comparability to earlier work. All other sentences (59,622 sentences with 1,032,091 words) were used as the training set. For each word in the training set, the local context was extracted and expressed in our supertag notation. The word/supertag pairs were then used to train the statistical part-of-speech tagger TnT (Brants, 2000), which performs trigram tagging efficiently and allows easy retraining on different data. However, a few of TnT’s limitations had to be worked around: since it cannot deal with words that have more than 510 different possible tags, we systematically replaced the rarest tags in the training set with a generic ‘OTHER’ tag until the limit was met. Also, in tagging mode it can fail to process sentences with many unknown words in close succession. In such cases, we simply ran it on shorter fragments of the sentence until no error occurred. Fewer than 0.5% of all sentences were affected by this problem even with the largest tag set.

A more serious problem arises when using a stochastic process to assign tags that partially predict structure: the tags emitted by the model may contradict each other. Consider, for instance, the following supertagger output for the previous example sentence:

```
es: +EXPL/R+   mag: +S/N+AUX, SUBJC
sein: PRED+AUX/L+   ...
```

The supertagger correctly predicts that the first three labels are EXPL, S, and AUX. It also predicts that the word ‘sein’ has a preceding PRED complement, but this is impossible if the two preceding words are labelled EXPL and S. Such contradictory information is not fatal in a robust system, but it is likely to cause unnecessary work for the parser when some rules demand the impossible. We therefore decided simply to ignore context predictions when they contradict the basic label predictions made for the same sentence; in other words, we pretend that the prediction for the third word was just +AUX/L+ rather than PRED+AUX/L+. Up to 13% of all predictions were simplified in this way for the most complex supertag model.

The last columns of Table 2 give the number of different supertags in the training set and the performance of the retrained TnT on the test set in single-tagging mode. Although the number of oc-

curing tags rises and the prediction accuracy falls with the supertag complexity, the correlation is not absolute: It seems markedly easier to predict supertags with complements but no direction information (C) than supertags with direction information but no complements (B), although the tag set is larger by an order of magnitude. In fact, the prediction of attachment direction seems much more difficult than that of undirected supertags in every case, due to the semi-free word order of German. The greater tag set size when predicting complements of each words is at least partly offset by the contextual information available to the n -gram model, since it is much more likely that a word will have, e.g., a ‘SUBJ’ complement when an adjacent ‘SUBJ’ supertag is present.

For the simplest model A, all 35 possible supertags actually occur, while in the most complicated model J, only 12,947 different supertags are observed in the training data (out of a theoretically possible 10^{24} for a set of 35 edge labels). Note that this is still considerably larger than most other reported supertag sets. The prediction quality falls to rather low values with the more complicated models; however, our goal in this paper is not to optimize the supertagger, but to estimate the effect that an imperfect one has on an existing parser. Altogether most results fall into a range of 70–80% of accuracy; as we will see later, this is in fact enough to provide a benefit to automatic parsing.

Although supertag accuracy is usually determined by simply counting matching and non-matching predictions, a more accurate measure should take into account how many of the individual predictions that are combined into a supertag are correct or wrong. For instance, a word that is attached to its left as a subject, is preceded by a preposition and an attributive adjective, and followed by an apposition would bear the supertag `PP, ATTR+SUBJ/L+APP`. Since the prepositional attachment is notoriously difficult to predict, a supertagger might miss it and emit the slightly different tag `ATTR+SUBJ/L+APP`. Although this supertag is technically wrong, it is in fact much more right than wrong: of the four predictions of label, direction, preceding and following dependents, three are correct and only one is wrong. We therefore define the *component accuracy* for a given model as the ratio of correct predictions among the possible ones, which results in a value of 0.75 rather than 0 for the exam-

ple prediction. The component accuracy of the supertag model J e. g. is in fact 84.5% rather than 67.6%. We would expect the component accuracy to match the effect on parsing more closely than the supertag accuracy.

3 Using supertag information in WCDG

Weighted Constraint Dependency Grammar (WCDG) is a formalism in which declarative constraints can be formulated that describe well-formed dependency trees in a particular natural language. A grammar composed of such constraints can be used for parsing by feeding it to a constraint-solving component that searches for structures that satisfy the constraints.

Each constraint carries a numeric score or *penalty* between 0 and 1 that indicates its importance. The penalties of all instances of constraint violations are multiplied to yield a score for an entire analysis; hence, an analysis that satisfies all rules of the WCDG bears the score 1, while lower values indicate small or large aberrations from the language norm. A constraint penalty of 0, then, corresponds to a hard constraint, since every analysis that violates such a constraint will always bear the worst possible score of 0. This means that of two constraints, the one with the *lower* penalty is more important to the grammar.

Since constraints can be soft as well as hard, parsing in the WCDG formalism amounts to multi-dimensional optimization. Of two possible analyses of an utterance, the one that satisfies more (or more important) constraints is always preferred. All knowledge about grammatical rules is encoded in the constraints that (together with the lexicon) constitute the grammar. Adding a constraint which is sensitive to supertag predictions will therefore change the objective function of the optimization problem, hopefully leading to a higher share of correct attachments. Details about the WDCG parser can be found in (Foth and Menzel, 2006).

A grammar of German is available (Foth et al., 2004) that achieves a good accuracy on written German input. Despite its good results, it seems probable that the information provided by a supertag prediction component could improve the accuracy further. First, because the optimization problem that WCDG defines is infeasible to solve exactly, the parser must usually use incomplete,

heuristic algorithms to try to compute the optimal analysis. This means that it sometimes fails to find the correct analysis even if the language model accurately defines it, because of search errors during heuristic optimization. A component that makes specific predictions about local structure could guide the process so that the correct alternative is tried first in more cases, and help prevent such search errors. Second, the existing grammar rules deal mainly with structural compatibility, while supertagging exploits patterns in the sequence of words in its input, i. e. both models contribute complementary information. Moreover, the parser can be expected to profit from supertags providing highly lexicalized pieces of information.

Model	Supertag accuracy	Component accuracy	Parsing accuracy	
			unlabelled	labelled
baseline	–	–	89.6%	87.9%
A	84.1%	84.1%	90.8%	89.4%
B	78.9%	85.7%	90.6%	89.2%
C	81.1%	88.5%	91.0%	89.6%
D	76.9%	90.8%	91.1%	89.8%
E	80.6%	91.8%	90.9%	89.6%
F	76.2%	90.9%	91.4%	90.0%
G	71.8%	81.3%	90.8%	89.4%
H	67.9%	85.8%	90.8%	89.4%
I	71.6%	84.3%	91.8%	90.4%
J	67.6%	84.5%	91.8%	90.5%

Table 3: Influence of supertag integration on parsing accuracy.

Constraint penalty	Parsing accuracy	
	unlabelled	labelled
0.0	3.7%	3.7%
0.05	85.2%	83.5%
0.1	87.6%	85.7%
0.2	88.9%	87.3%
0.5	91.2%	89.5%
0.7	91.5%	90.1%
0.9	91.8%	90.5%
0.95	91.1%	89.8%
1.0	89.6%	87.9%

Table 4: Parsing accuracy depending on different strength of supertag integration.

To make the information from the supertag sequence available to the parser, we treat the complex supertags as a set of predictions and write constraints to prefer those analyses that satisfy them. The predictions of label and direction made by models A and B are mapped onto two constraints which demand that each word in the analysis should exhibit the predicted label and direction. The more complicated supertag models constrain the local context of each word further. Effectively, they predict that the specified dependents of

a word occur, and that no other dependents occur. The former prediction equates to an existence condition, so constraints are added which demand the presence of the predicted relation types under that word (one for left dependents and one for right dependents). The latter prediction disallows all other dependents; it is implemented by two constraints that test the edge label of each word-to-word attachment against the set of predicted dependents of the regent (again, separately for left and right dependents). Altogether six new constraints are added to the grammar which refer to the output of the supertagger on the current sentence.

Note that in contrast to most other approaches we do not perform multi-supertagging; exactly one supertag is assumed for each word. Alternatives could be integrated by computing the logical disjunctions of the predictions made by each supertag, and then adapting the new constraints accordingly.

4 Experiments

We tested the effect of supertag predictions on a full parser by adding the new constraints to the WCDG of German described in (Foth et al., 2004) and re-parsing the same 1,000 sentences from the NEGRA corpus. The quality of a dependency parser such as this can be measured as the ratio of correctly attached words to all words (structural accuracy) or the ratio of the correctly attached and correctly labelled words to all words (labelled accuracy). Note that because the parser always finds exactly one analysis with exactly one subordination per word, there is no distinction between recall and precision. The structural accuracy without any supertags is 89.6%.

To determine the best trade-off between complexity and prediction quality, we tested all 10 supertag models against the baseline case of no supertags at all. The results are given in Table 3. Two observations can be made about the effect of the supertag model on parsing. Firstly, all types of supertag prediction, even the very basic model A which predicts only edge labels, improve the overall accuracy of parsing, although the baseline is already quite high. Second, the richer models of supertags appear to be more suitable for guiding the parser than the simpler ones, even though their own accuracy is markedly lower; almost one third of the supertag predictions according to the most compli-

cated definition J are wrong, but nevertheless their inclusion reduces the remaining error rate of the parser by over 20%.

This result confirms the assumption that if supertags are integrated as individual constraints, their component accuracy is more important than the supertag accuracy. The decreasing accuracy of more complex supertags is more than counterbalanced by the additional information that they contribute to the analysis. Obviously, this trend cannot continue indefinitely; a supertag definition that predicted even larger parts of the dependency tree would certainly lead to much lower accuracy by even the most lenient measure, and a prediction that is mostly wrong must ultimately degrade parsing performance. Since the most complex model J shows no parsing improvement over its successor I, this point might already have been reached.

The use of supertags in WCDG is comparable to previous work which integrated POS tagging and chunk parsing. (Foth and Hagenström, 2002; Daum et al., 2003) showed that the correct balance between the new knowledge and the existing grammar is crucial for successful integration. This is achieved by means of an additional parameter, modeling how trustworthy supertag predictions are considered. Its effect is shown in Table 4. As expected, making supertag constraints hard (with a value of 0.0) over-constrains most parsing problems, so that hardly any analyses can be computed. Other values near 0 avoid this problem but still lead to much worse overall performance, as wrong or even impossible predictions too often overrule the normal syntax constraints. The previously used value of 0.9 actually yields the best results with this particular grammar.

The fact that a statistical model can improve parsing performance when superimposed on a sophisticated hand-written grammar is of particular interest because the statistical model we used is so simple, and in fact not particularly accurate; it certainly does not represent the state of the art in supertagging. This gives rise to the hope that as better supertaggers for German become available, parsing results will continue to see additional improvements, i.e., future supertagging research will directly benefit parsing. The obvious question is how great this benefit might conceivably become under optimal conditions. To obtain this upper limit of the utility of supertags we repeated

Supertag model	Constraint penalty	
	0.9	0.0
A	92.7% / 92.2%	94.0% / 94.0%
B	94.3% / 93.7%	96.0% / 96.0%
C	92.8% / 92.4%	94.1% / 94.1%
D	94.3% / 93.8%	96.0% / 96.0%
E	93.1% / 92.6%	94.3% / 94.3%
F	94.6% / 94.1%	96.1% / 96.1%
G	94.2% / 93.7%	95.8% / 95.8%
H	95.2% / 94.7%	97.4% / 97.4%
I	97.1% / 96.8%	99.5% / 99.5%
J	97.1% / 96.8%	99.6% / 99.6%

Table 5: Unlabelled and labelled parsing accuracy with a simulated perfect supertagger.

the process of translating each supertag into additional WCDG constraints, but this time using the test set itself rather than TnT’s predictions.

Table 5 again gives the unlabelled and labelled parsing accuracy for all 10 different supertag models with the integration strengths of 0 and 0.9. (Note that since all our models predict the edge label of each word, hard integration of perfect predictions eliminates the difference between labelled and unlabelled accuracy.) As expected, an improved accuracy of supertagging would lead to improved parsing accuracy in each case. In fact, knowing the correct supertag would solve the parsing problem almost completely with the more complex models. This confirms earlier findings for English (Nasr and Rambow, 2004).

Since perfect supertaggers are not available, we have to make do with the imperfect ones that do exist. One method of avoiding some errors introduced by supertagging would be to reject supertag predictions that tend to be wrong. To this end, we ran the supertagger on its training set and determined the average component accuracy of each occurring supertag. The supertags whose average precision fell below a variable threshold were not considered during parsing as if the supertagger had not made a prediction. This means that a threshold of 100% corresponds to the baseline of not using supertags at all, while a threshold of 0% prunes nothing, so that these two cases duplicate the first and last line from Table 2.

As Table 6 shows, pruning supertags that are wrong more often than they are right results in a further small improvement in parsing accuracy: unlabelled syntax accuracy rises up to 92.1% against the 91.8% if all supertags of model J are used. However, the effect is not very noticeable, so that it would be almost certainly more useful to

Threshold	Parsing accuracy	
	unlabelled	labelled
0%	91.8%	90.5%
20%	91.8%	90.4%
40%	91.9%	90.5%
50%	92.0%	90.7%
60%	92.1%	91.0%
80%	91.4%	90.0%
100%	89.6%	87.9%

Table 6: Parsing accuracy with empirically pruned supertag predictions.

improve the supertagger itself rather than second-guess its output.

5 Related work

Supertagging was originally suggested as a method to reduce lexical ambiguity, and thereby the amount of disambiguation work done by the parser. Sakar et al. (2000) report that this increases the speed of their LTAG parser by a factor of 26 (from 548k to 21k seconds) but at the price of only being able to parse 59% of the sentences in their test data (of 2250 sentences), because too often the correct supertag is missing from the output of the supertagger. Chen et al. (2002) investigate different supertagging methods as pre-processors to a Tree-Adjoining Grammar parser, and they claim a 1-best supertagging accuracy of 81.47%, and a 4-best accuracy of 91.41%. With the latter they reach the highest parser coverage, about three quarters of the 1700 sentences in their test data.

Clark and Curran (2004a; 2004b) describe a combination of supertagger and parser for parsing Combinatory Categorical Grammar, where the tagger is used to filter the parses produced by the grammar, before the computation of the model parameters. The parser uses an incremental method: the supertagger first assigns a small number of categories to each word, and the parser requests more alternatives only if the analysis fails. They report 91.4% precision and 91.0% recall of unlabelled dependencies and a speed of 1.6 minutes to parse 2401 sentences, and claim a parser speedup of a factor of 77 thanks to supertagging.

The supertagging approach that is closest to ours in terms of linguistic representations is probably (Wang and Harper, 2002; Wang and Harper, 2004) whose ‘Super Abstract Role Values’ are very similar to our model F supertags (Table 2). It is interesting to note that they only report between 328 and 791 SuperARVs for different corpora, whereas

we have 2026 category F supertags. Part of the difference is explained by our larger label set: 35, the same as the number of model A supertags in table 2 against their 24 (White, 2000, p. 50). Also, we are not using the same corpus. In addition to determining the optimal SuperARV sequence in isolation, Wang and Harper (2002) also combine the SuperARV n -gram probabilities with a dependency assignment probability into a dependency parser for English. A maximum tagging accuracy of 96.3% (for sentences up to 100 words) is achieved using a 4-gram n -best tagger producing the 100 best SuperARV sequences for a sentence. The tightly integrated model is able to determine 96.6% of SuperARVs correctly. The parser itself reaches a labelled precision of 92.6% and a labelled recall of 92.2% (Wang and Harper, 2004).

In general, the effect of supertagging in the other systems mentioned here is to reduce the ambiguity in the input to the parser and thereby increase its speed, in some cases dramatically. For us, supertagging decreases the speed slightly, because additional constraints means more work for the parser, and because our supertagger-parser integration is not yet optimal. On the other hand it gives us better parsing accuracy. Using a constraint penalty of 0.0 for the supertagger integration (c.f. Table 5) does speed up our parser several times, but would only be practical with very high tagging accuracy. An important point is that for some other systems, like (Sarkar et al., 2000) and (Chen et al., 2002), parsing is not actually feasible without the supertagging speedup.

6 Conclusions and future work

We have shown that a statistical supertagging component can significantly improve the parsing accuracy of a general-purpose dependency parser for German. The error rate among syntactic attachments can be reduced by 24% over an already competitive baseline. After all, the integration of the supertagging results helped to reach a quality level which compares favourably with the state-of-the-art in probabilistic dependency parsing for German as defined with 87.34%/90.38% labelled/unlabelled attachment accuracy on this years shared CoNLL task by (McDonald et al., 2005) (see (Foth and Menzel, 2006) for a more detailed comparison). Although the statistical model used in our system is rather simple-minded, it clearly captures at least some distributional char-

acteristics of German text that the hand-written rules do not.

A crucial factor for success is the defeasible integration of the supertagging predictions via soft constraints. Rather than pursuing a strict filtering approach where supertagging errors are partially compensated by an n -best selection, we commit to only one supertag per word, but reduce its influence. Treating supertag predictions as weak preferences yields the best results. By measuring the accuracy of the different types of predictions made by complex supertags, different weights could also be assigned to the six new constraints.

Of the investigated supertag models, the most complex ones guide the parser best, although their own accuracy is not the best one, even when measured by the more pertinent component accuracy. Since purely statistical parsing methods do not reach comparable parsing accuracy on the same data, we assume that this trend does not continue indefinitely, but would stop at some point, perhaps already reached.

References

- S. Bangalore and A. K. Joshi. 1999. Supertagging: an approach to almost parsing. *Computational Linguistics*, 25(2):237–265.
- T. Brants, R. Hendriks, S. Kramp, B. Krenn, C. Preis, W. Skut, and H. Uszkoreit. 1997. Das NEGRA-Annotationsschema. Technical report, Universität des Saarlandes, Computerlinguistik.
- S. Brants, St. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER treebank. In *Proc. Workshop on Treebanks and Linguistic Theories*, Sozopol.
- T. Brants. 2000. TnT – A statistical part-of-speech tagger. In *Proc. the 6th Conf. on Applied Natural Language Processing, ANLP-2000*, pages 224–231, Seattle, WA.
- J. Chen, S. Bangalore, M. Collins, and O. Rambow. 2002. Reranking an N-gram supertagger. In *Proc. 6th Int. Workshop on Tree Adjoining Grammar and Related Frameworks*.
- S. Clark and J. R. Curran. 2004a. The importance of supertagging for wide-coverage CCG parsing. In *Proc. 20th Int. Conf. on Computational Linguistics*.
- S. Clark and J. R. Curran. 2004b. Parsing the WSJ using CCG and log-linear models. In *Proc. 42nd Meeting of the ACL*.
- M. Daum, K. Foth, and W. Menzel. 2003. Constraint based integration of deep and shallow parsing techniques. In *Proc. 11th Conf. of the EACL*, Budapest, Hungary.
- M. Daum, K. Foth, and W. Menzel. 2004. Automatic transformation of phrase treebanks to dependency trees. In *Proc. 4th Int. Conf. on Language Resources and Evaluation, LREC-2004*, pages 99–106, Lisbon, Portugal.
- K. Foth and J. Hagenström. 2002. Tagging for robust parsers. In *2nd Workshop on Robust Methods in Analysis of Natural Language Data, ROMAND-2002*, pages 21 – 32, Frascati, Italy.
- K. Foth and W. Menzel. 2006. Hybrid parsing: Using probabilistic models as predictors for a symbolic parser. In *Proc. 21st Int. Conf. on Computational Linguistics, Coling-ACL-2006*, Sydney.
- K. Foth, M. Daum, and W. Menzel. 2004. A broad-coverage parser for german based on defeasible constraints. In *7. Konferenz zur Verarbeitung natürlicher Sprache, KONVENS-2004*, pages 45–52, Wien.
- R. McDonald, F. Pereira, K. Ribarov, and J. Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proc. Human Language Technology Conference, HLT/EMNLP-2005*, Vancouver, B.C.
- A. Nasr and O. Rambow. 2004. A simple string-rewriting formalism for dependency grammar. In *Coling-Workshop Recent Advances in Dependency Grammar*, pages 17–24, Geneva, Switzerland.
- A. Sarkar, F. Xia, and A. Joshi. 2000. Some experiments on indicators of parsing complexity for lexicalized grammars. In *Proc. COLING Workshop on Efficiency in Large-Scale Parsing Systems*.
- Y. Schabes and A. K. Joshi. 1991. Parsing with lexicalized tree adjoining grammar. In M. Tomita, editor, *Current Issues in Parsing Technologies*. Kluwer Academic Publishers.
- I. Schröder, W. Menzel, K. Foth, and M. Schulz. 2000. Modeling dependency grammar with restricted constraints. *Traitement Automatique des Langues (T.A.L.)*, 41(1):97–126.
- W. Wang and M. P. Harper. 2002. The SuperARV language model: Investigating the effectiveness of tightly integrating multiple knowledge sources. In *Proc. Conf. on Empirical Methods in Natural Language Processing, EMNLP-2002*, pages 238–247, Philadelphia, PA.
- W. Wang and M. P. Harper. 2004. A statistical constraint dependency grammar (CDG) parser. In *Proc. ACL Workshop Incremental Parsing: Bringing Engineering and Cognition Together*, pages 42–49, Barcelona, Spain.
- Ch. M. White. 2000. *Rapid Grammar Development and Parsing: Constraint Dependency Grammar with Abstract Role Values*. Ph.D. thesis, Purdue University, West Lafayette, IN.