N-gram and PCFG Models for Morpho-Syntactic Tagging of German*

Erhard W. Hinrichs and Julia S. Trushkina

Eberhard-Karls-Universität Tübingen
Seminar für Sprachwissenschaft, Computerlinguistik
Wilhelmstr. 19, 72074 Tübingen, Germany
email: {eh,jul}@sfs.uni-tuebingen.de

1 Introduction

The purpose of this paper is to investigate alternative methods of automatic, morpho-syntactic annotation with a large tagset that can be effectively trained on manually annotated data of moderate size. We will argue that standard n-gram taggers such as TnT are inadequate for this task, but that probabilistic context-free grammars provide a suitable alternative that yields acceptable results even for moderate amounts of training data.
The utility of treebanks is directly proportional to the richness of linguistic annotation reflected in the trees. For languages with rich inflectional morphology such as German, case, gender, person and number information on lexical categories is of crucial importance and has, therefore, been incorporated into the NEGRA (Skut et al., 1997) and TüBa-D (Hinrichs et al., 2000) treebanks. Consequently these treebanks provide a much more fine-grained inventory of morpho-syntactic tags compared to the STTS tagset (Schiller et al., 1995) , the widely accepted inventory of part-of-speech categories for German.
While such richer annotation schemes are desirable from a descriptive and an application-oriented perspective, they lead to an increase in the amount of labor for human annotators. The need for robust semi-automatic or automatic annotation

*We wish to thank Thorsten Brants and Tylman Ule for valuable comments related to this paper. In particular, we are indebted to Tylman Ule whose own PCFG research reported in this volume provided the inspiration for the tree transformations discussed in section 3.3.

tools which can aid or replace human annotation, therefore, becomes even more important. For part-of-speech annotation, probabilistic taggers such as TnT provide a tool of this kind. One limitation of such taggers lies in the size of the tagset used in the training corpus. As the size of the tagset increases by the introduction of morpho-syntactic categories, tagger performance typically decreases.
As (Brants, 2000) has shown, the TnT tagger achieves 96.70% accuracy when trained on a newspaper corpus of 300.000 tokens with the STTS tagset that comprises 54 basic part-of-speech tags for German. Adding morpho-syntactic features such as case, number, person, gender, tense and mood to the basic STTS labels results in a set of 718 possible tags.

2 TnT experiments

The TnT tagger was trained with this enlarged tagset on 51288 manually annotated tokens from the `taz` newspaper portion of the TüBa-D treebank (taz, 1999). Additional 5854 lexical tokens served as the development set. When tested on 8949 tokens from the same corpus, the TnT tagger achieves a baseline accuracy of 74.97%. A first improvement to 80.61% can be achieved if unknown words in the test corpus are tagged not via the TnT guesser, but by a back-up lexicon containing the set of possible morpho-syntactic analyses for each unknown word. Further improvements can be obtained by either increasing the size of the training data or by reducing the tagset. Considering the amount of manual effort necessary, the former strategy is not feasible in practice. The latter strategy has been advocated by (Tufiş, 2000) and by (Dienes and Oravecz, 2000).

2.1 Reduced tagset approaches

In the approaches of (Tufiş, 2000) and (Dienes and Oravecz, 2000) a reduced tagset is designed for an intermediary tagging with the following replacement of the tags from the reduced tagset with the more informative tags of the original large tagset. The two approaches differ in the methods used for tagset compaction. Tufiş' tiered tagging (T-tagging) approach successively reduces the size of the original tagset (allowing for a 10% loss of information) while Dienes and Oravecz' bottom-up tagset design (BUTD) maximally reduces the original tagset without loss of information and then re-introduces morpho-syntactic features to expand the reduced tagset to a set of tags that exhibit sufficient distributional cues for the tagger.
Figure 1 shows the results of applying these two algorithms to the German data. The BUTD approach fails to reach the baseline set by TnT with the maximally reduced tagset (72.33%). Experiments with expanded tagsets whose classes respect

	development data		test data	
	pure	back-up	pure	back-up
TnT	75.09	80.74	74.97	80.61
BUTD				
straightforward	72.51	72.53	72.15	72.33
expanded tagsets	76.23	78.49	75.43	78.19
T-tagging				
straightforward (10% info loss)	82.71	83.58	81.84	83.13
Case, Number and Person only	83.59	83.02	83.26	83.01
no Case, Number and Person	85.39	85.58	84.03	84.94

Figure 1: TnT experiments

differences in major word classes yield a maximal, but still below baseline, result of 78.19%.
The T-tagging approach is able to outperform the TnT baseline. The standard margin of 10% information loss set by Tufiş' algorithm yields 83.13% for German data. However, this higher result is achieved at the cost of losing information contained in the original tagset. The algorithm described in (Tufiş, 2000) allows to leave out an attribute from all the tags in the reduced tagset if the elimination results in limited percentage (less than 10%) of words becoming ambiguous after recovering the original tags. The recovering process is lexicon driven: it is equivalent to computing the intersection between the set of original tags that map into the reduced tag assigned to the token and the set of analyses specified for the token in the lexicon. For resolving the remaining 10% ambiguity, the algorithm relies on hand-crafted rules. However, for German this strategy turns out not to work: 10% information loss results in the elimination of all morphological features on major word classes, e.g. finite auxiliary verbs, articles, proper nouns as well as personal, demonstrative and relative pronouns and determiners. This type of information is not recoverable by hand-crafted rules, thus limiting the utility of the tagged output.
It is instructive to profile the T-tagging approach by comparing the retention of different sets of crucial morphological features. If only case and number information is retained on nominal categories and only number and person information is retained on finite verbs, this results in a lower accuracy (83.01%) than if only these features are deleted from the tagset and all other morphological features are retained on these and on all other word classes (84.94%). Case, number, and person are generally regarded as the most basic morphological features to be included in any tagset with inflectional information. The fact that this feature set is more

difficult to accommodate than all other morphological features combined points at a crucial deficiency in the underlying model for the task at hand.

2.2 Limitations of N-gram models

N-gram taggers such as TnT consider only sequences of **n** words and their candidate tags, i.e. very local contexts, as the basis for determining the most likely sequence of tags for the sentence. This Markovian assumption proves harmful for decisions that crucially require larger context windows. Case, person and number information is precisely of this nature, since successful disambiguation needs to rely on genuinely syntactic phenomena such as subject-verb agreement, valency of main verbs, and morphological features of other nominal elements in the sentence. While at first sight it seems that case and nominal number can be resolved within a small context window via phrase-internal agreement, the findings of (Hinrichs and Trushkina, 2002) show that in a lot of cases use of syntactic heuristics which take into account distant features is necessary. The most prominent example of this kind is illustrated in (1). The example is taken from the error analysis of TnT performance on test data.

(1)	Die	Frage	nach	der	Form	beantwortet	er	dann	auch	so:
	The	question	about	the	form	answers	he	then	also	in this way

‘He answers the question about the form in this way:’

The sentence contains an unambiguous nominative pronoun *er* and a noun phrase *die Frage*. The latter NP represents a common nominative-accusative ambiguity. TnT wrongly assigned nominative case to the tokens *die Frage* even though the combination of the nominative and accusative NPs in a clause is much more likely than a pattern of two nominative NPs. This deficiency in the statistical model used by the TnT tagger is due to its extremely local context window.
The most widely used probabilistic models that can incorporate more global structural information are probabilistic context-free grammars (PCFGs). The conjecture that PCFGs yield better results is confirmed by experiments with the PCFG-parser LoPar (Schmid, 2000). These experiments are described in the following section.

3 PCFG experiments

3.1 Tagging mode of LoPar

LoPar is a particularly good tool to use since it provides dedicated Viterbi parsing and tagging modes. The corresponding modes give different results when applied to the task of morpho-syntactic annotation with large tagsets. In Viterbi parsing mode, LoPar computes the most probable tree structure for a given input string. This mode can be used for part-of-speech tagging by outputting the sequence of part-of-speech tags that appear as preterminal nodes of this most likely tree. In tagging mode, the best tag sequence is computed independently of the most likely parse tree. In this mode the best tag sequence is defined as the sequence of those tags that yield the maximal product of the inside and outside probabilities among the candidate tags for a given word. A more formal comparison between the two modes is given in Appendix A.

Since LoPar's tagging mode computes the most likely tag sequence independently of the most likely parse tree, the best tag for a given word will be identified as that tag which provides the best balance between the likelihood of the tag itself and the likelihood of the surrounding syntactic context. Since this choice is determined for each tag independently of all the others, the resulting tag sequence may well differ from the tag sequence of the most likely parse tree. Viterbi parsing has as its goal the construction of a single tree that provides the best balance between the individual probabilities of the candidate structures of its parts. In order to arrive at this balance, the most likely analysis of an individual phrase or word may not be incorporated into the most likely tree if the outside probability for that analysis is much inferior to the outside probabilities of other candidate analyses for the same word or phrase. Thus, Viterbi parsing optimizes global structure, not individual substructures. Since part-of-speech tagging is of a more local nature compared to full sentential analysis, it should therefore be expected that independent maximization of tags for individual words should perform better than global maximization of full sentential structures.

The experiments with both Viterbi parsing and tagging modes demonstrate an average improvement of the tagging mode performance over the performance of the parsing mode that amounts to 0.9%. This difference in results stresses the independent nature of the tagging task in PCFG application. Tagging is viewed not as a subtask of full parsing, but as a separate problem and represents a challenge of a different kind for a PCFG. What is usually assumed as input for parsing – a sequence of part-of-speech tags – is to be found in the tagging problem, and the main emphasis is put on identification of those regularities of syntactic structure that are highly predictive for the assignment of the correct tag sequence.

Training, development and test material for LoPar consisted of treebank data for the same manually annotated tokens used in the TnT experiments. The best results obtained by LoPar yield an accuracy of 87.34% on the test data and of 88.21% on the development set. These results required redefinition of non-terminal nodes and tree transformations compared to the original treebank. This confirms the findings of (Johnson, 1998) who showed that the performance of PCFG parsing varies considerably depending on the choice of non-terminal symbols and the tree trans-

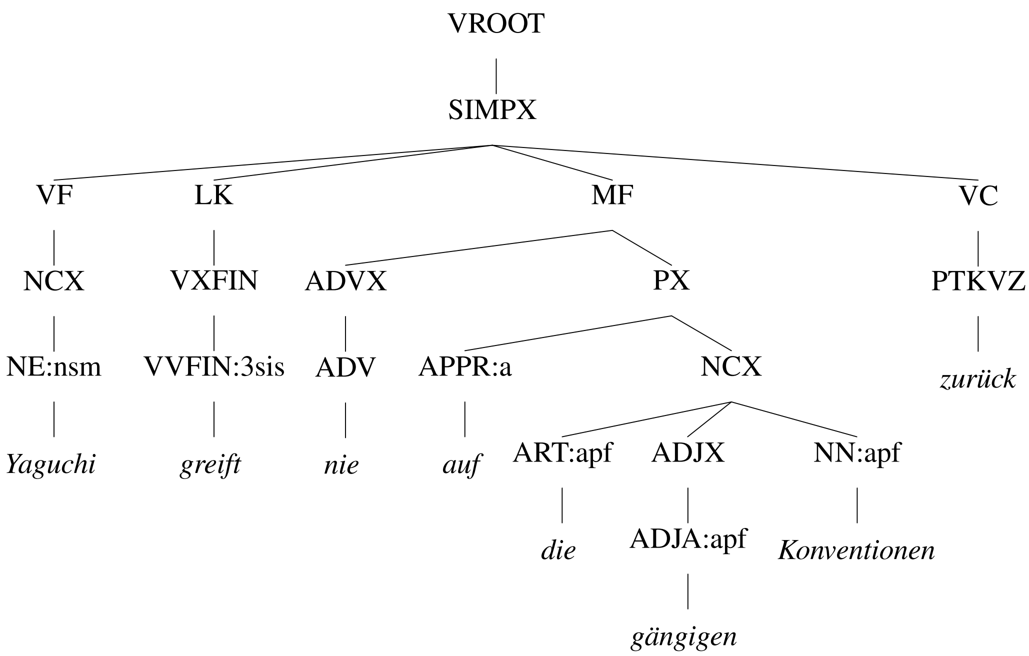

Figure 2: Example of an original tree

formations performed on the Penn treebank.

3.2 Original treebank structures

The tree in Figure 2 exemplifies the type of structures and categories used in the baseline LoPar experiment. The tree structure represents the label bracketing of the original TüBa-D treebank format but leaves out function argument information represented as edge labels in the TüBa-D treebank.
Treebank segmentation units may contain more than one sentence and/or phrase, which results in disconnected trees in some cases. To avoid disconnected trees in the LoPar input, a root node `VROOT` was added to all treebank segmentation units.
The framework of topological fields (Höhle, 1985) that helps to capture fundamental word order regularities of German sentence structure is adopted in the treebank. Node labels `VF`, `LK`, `MF` and `VC` represent such topological fields: sentence brackets `LK` and `VC` formed by verbal elements divide a main clause into a *Vorfeld* (`VF` - initial field), a *Mittelfeld* (`MF` - middle field) and a *Nachfeld* (`NF` - final field).
Maximal phrasal categories can be identified by the character `X` in the node label, preceded by the abbreviation for the type of phrase. E.g. `PX` and `NCX` are the TüBa-D treebank equivalents of the more traditional labels `PP` and `NP`.

		precision	# of unparsed sentences
1.	baseline:	66.40	5
2.	back-up lexicon added	77.65	5
3.	topological fields (except for VC and C) deleted	77.58	5
4.	case passed up to NX and NCX	84.23	6
5.	grammatical functions (-ON and -OA) added and passed up to SIMPX + rules binarized	84.98	5
6.	morph. info passed up to NXs and VXFINs	87.62	7
7.	FIN label with number passed up to SIMPX	**88.21**	9
8.	results on test data	**87.69**	11

Figure 3: LoPar PCFG experiments

Preterminal nodes are formed by STTS tags and morphological codes separated by the colon delimiter. Morphological codes of nouns, adjectives and articles represent values for case (first slot), number (middle slot) and gender (last slot). Verbal inflectional morphology - person, number, mood and tense - is encoded in the corresponding four slots in finite verb tags. For prepositions the only morphological feature present is a value of the required by the preposition case.
Figure 3 summarizes the LoPar experiments on the development set and the final result on the test data. The baseline of 66.40% is obtained by training LoPar on the treebank as is. Adding a back-up lexicon with candidate morpho-syntactic analyses for unknown words improves accuracy to 77.65%. Further improvements are obtained by enriching the set of non-terminal nodes and by tree transformations. These improvements are summarized in lines 3 to 7 of Figure 3.

3.3 Treebank transformations

Enriching the set of non-terminal nodes and tree transformations both have the goal of optimizing the treebank structure for training a model that best predicts the correct morpho-syntactic tag sequence.
Enriching the set of non-terminal nodes aims at weakening the independence assumption inherent in PCFGs. Percolation of relevant morphological and functional information between lexical and phrasal nodes allows the model to better capture regularities in syntactic structure for correct assignment of tags:
1. Passing morphological information (case, number and gender) on phrasal categories that constitute an NP (nouns, adjectives, determiners etc.) facilitates the recognition of intra-phrasal agreement.
Consider as an example the tree in Figure 5. If no morphological information is en-

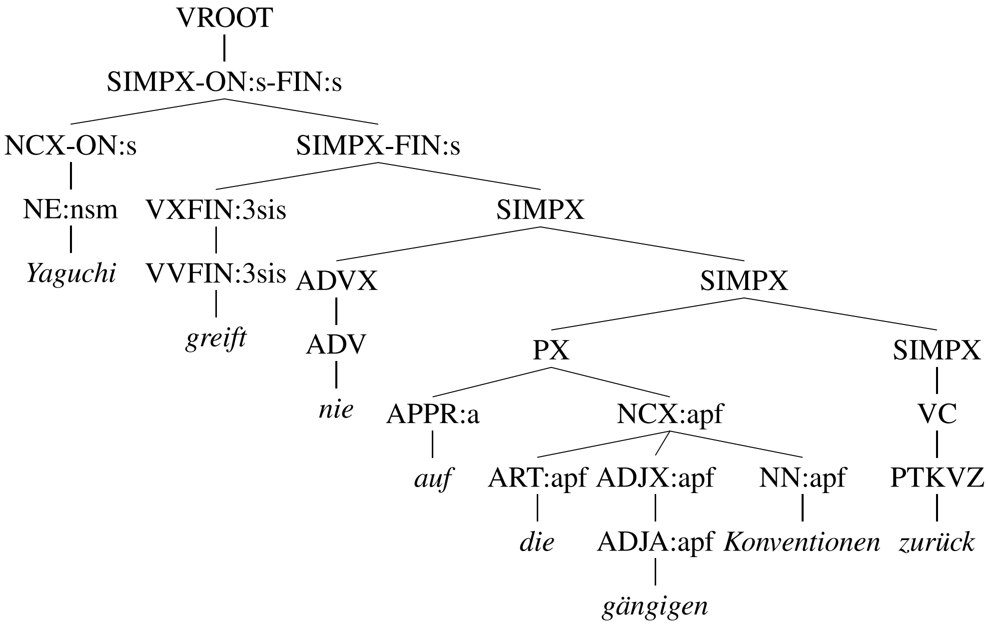

Figure 4: Example of a transformed tree

coded on the node ADJX, a formation of a subtree with accusative article and noun and a nominative adjective would be possible, since expansion of the adjectival phrase is independent of its sisters. Percolating morphological information from the adjective to its mother node makes the structure more rigid: the probability of a noun phrase expanding to categories with different morphological values is very low, as well as the probability of expanding an accusative ADJX to a non-accusative adjective.

2. Percolation of function labels -OA and -ON (accusative object and subject) up to SIMPX (clause) nodes helps to prevent formation of two-subject as well as subject-less clauses, which are less frequent in the treebank.[1] In a way this information reflects the subcategorization frame of the verb. It is interesting to note, though, that inclusion of dative and genitive objects in the treebank structure hurts tagger performance.

3. Passing the FIN-label of finite verbs up to SIMPX nodes allows to trace regularities between finiteness of a clause and morphological cases of NPs in a clause: a finite verb in a clause requires presence of the subject, whereas absence of finite verb is more probable for subject-less clauses.

[1]These function labels are present in the TüBa-D treebank and are encoded there as edge labels between tree nodes.

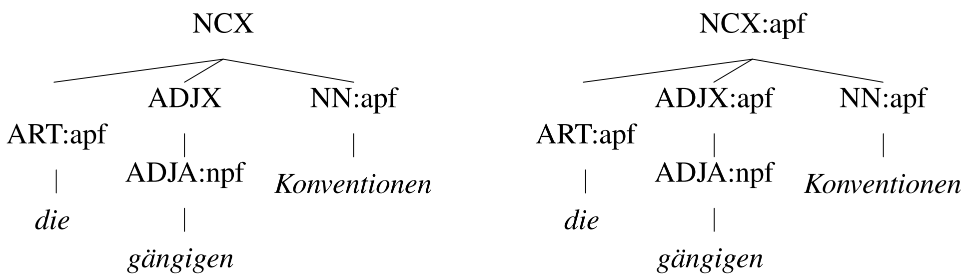

Figure 5: NP subtree without and with percolation of morphological features

4. Introducing a feature for number on -FIN and -ON nodes makes capturing of subject-verb agreement possible.
Both -ON and -FIN labels are passed to SIMPX nodes. Trained on a treebank with correct structures, the model will consider a clause where the subject and the finite verb have different number less probable than a clause in which the subject and the finite verb agree in number.
Transformation of trees aims at two goals: to facilitate tracing of the relevant information in the sentence structure and to overcome the data-sparseness problem created by encoding detailed information on the nodes.
1. Topological fields such as `VF`, `MF`, `NF` are present in the original treebank as an intermediate layer of syntactic structure. Elimination of all topological field information except `C` and `VC` proved advantageous since retention of only these two fields turned out to be sufficient for determining the syntactic macrostucture across different clause types. The `C` field (short for "complementizer field") and the -FIN marking on the verbal complex node reliably identify (verb final) subordinate clauses. By contrast, verb-first and verb-second clauses can be identified by the position of the finite verb (without retention of the `LK` topological field) together with the absence of a -FIN specification on the verbal complex.
2. Encoding morphological and functional information on the nodes leads to a severe increase in the size of the rule set with an accompanying drop in recall (i.e. an increase in the number of unparsed sentences). This can be counterbalanced by binarization of tree structures, which makes the grammar more flexible by allowing for structures not present in the training data to be created in the tagging phase.
Changes to the treebank were introduced in the order in which they are presented in Figure 3. Some of the changes improve performance only in conjunction with others, such as the cluster of transformations described in line 5 of Figure 3. These transformations include adding and percolating grammatical functions in conjunction with rule binarization. When introduced independently of the others, any of these three changes turn out to deteriorate performance.

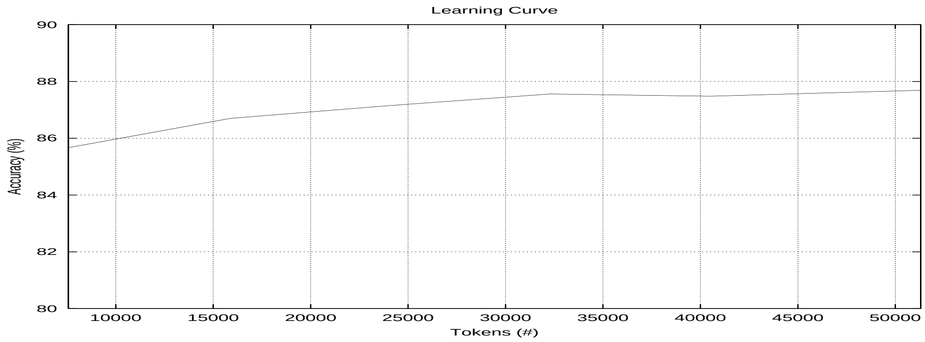

Figure 6: LoPar learning curve

The tree in Figure 4 exemplifies the type of structures and categories used in the transformed treebank.

3.4 Discussion

The best result obtained by the TnT models without loss of information in the underlying tagset has an accuracy of 80.61% and an error rate of 19.31%. The accuracy of 87.69% achieved by the LoPar model amounts to a significant reduction in the error rate of the best TnT model: a 36.51% reduction. Moreover, the accuracy is high enough to make the model useful for semi-automatic treebank annotation with expanded tagsets. If on average more than 8 out of 10 automatically assigned tags are correct, automatic annotation followed by manual error correction is more time-efficient compared to fully manual annotation. The PCFG model can also be used for the automatic error detection in manually annotated treebanks. If manually annotated data are used as test data, putative errors of the LoPar output can point to potential annotation errors in the gold standard.
Finally, note that the size of training corpus required to achieve acceptable results for the LoPar model is rather modest. (Brants, 2000) reports a learning curve for TnT models trained on the NEGRA corpus with the much smaller STTS tagset that flattens out only at 320K lexical tokens. The training corpus used in our experiments is little more than 1/6th in size, and, quite surprisingly, charting a learning curve for our training corpus already shows a significant flattening out. As noted earlier, richer annotation schemes lead to an increase in the amount of labor for human annotators, putting a natural bound on the size of resource that can be constructed by such means. If LoPar models can be trained successfully on manually annotated treebanks of such modest size, this highlights the feasibility and applicability of the overall approach.

References

Brants, T. TnT - A Statistical Part-of-Speech Tagger. *Applied Natural Language Processing Conference (ANLP 2000)*. 2000.

Dienes, P., Oravecz, C. Bottom-Up Tagset Design from Maximally Reduced Tagset. *International Conference on Computational Linguistics (COLING 2000)*, pages 42 – 47. 2000.

Hinrichs, E., J. Bartels, Y. Kawata, V. Kordoni, Telljohann, H. The Verbmobil Treebanks. In *5. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2000)*, pages 107 – 112. Ilmenau, 2000.

Hinrichs, E., Trushkina, J. Forging Agreement: Morphological Disambiguation of Noun Phrases. *First Workshop on Treebanks and Linguistic Theory (TLT 2002)*, pages 78 – 95. 2002. *http://www.BulTreeBank.org/Proceedings.html*.

Höhle, T. Der Begriff "Mittelfeld", Anmerkungen über die Theorie der topologischen Felder. *Akten des Siebten Internationalen Germanistenkongresses 1985*, pages 329–340. 1985.

Johnson, M. PCFG Models of Linguistic Tree Representations. *Computational Linguistics* 24(4), pages 613 – 632. 1998.

Manning, C., Schütze, H. Foundations of Statistical Natural Language Processing. *MIT Press*. 1999.

Schiller, A., Teufel, S., Thielen, C. Guidelines für das Tagging deutscher Textkorpora mit STTS. Universität Stuttgart and Universität Tübingen. 1995.

Schmid, H. LoPar: Design and Implementation. *Arbeitspapiere des Sonderforschungsbereiches* 340, No. 149, IMS Stuttgart. 2000.

Skut, W., Krenn B., Brants, T., Uszkoreit, H. An Annotation Scheme for Free Word Order Languages. *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*. 1997.

taz – die Tageszeitung (CD-ROM). September 1986 – May 1999. *www.taz.de*.

Tufiş, D. Using a Large Set of EAGLES-compliant Morpho-Syntactic Descriptors as a Tagset for Probabilistic Tagging. *International Conference on Language Resources and Evaluation (LREC 2000)*, pages 1105 – 1112. 2000.

Appendix A

As (Manning and Schütze, 1999) point out, the joint probability of a string $\mathtt{w}_{1m}$ and a node $\mathtt{N}_{pq}$ spanning from string position $\mathtt{p}$ to $\mathtt{q}$, given a grammar $\mathtt{G}$, can be computed by the formula in A1.

$$P(w_{1m}, N_{pq} \mid G) = \sum_j \alpha_j(p,q)\beta_j(p,q) \tag{A1}$$

$\alpha_j(p,q)$ denotes the outside probability of the node N_{pq} and $\beta_j(p,q)$ denotes its inside probability. For a definition of inside and outside probabilities see (Manning and Schütze, 1999).

In LoPar tagging mode the best tag sequence is defined as the sequence of those tags that yield the maximal product of the inside and outside probabilities among the candidate tags for a given word. Each tag is therefore computed by the following formula:

$$\arg\max_j \alpha_j(k,k)P(N^j \to w_k) \tag{A2}$$

A2 is an instantiation of A1 for preterminal nodes spanning from k to k with the sum operator replaced by $arg\ max$. $P(N^j \to w_k)$ denotes the inside probability of the tag.

In Viterbi parsing mode, LoPar computes the sequence of part-of-speech tags that appear as preterminal nodes of the most likely parse tree. What makes a direct comparison to the computation of tag sequence with the Viterbi parsing mode difficult is that computation of the most likely parse tree is usually performed in terms of inside probabilities only. This amounts to instantiating p and q in formula A1 to 1 and m and to a Viterbi-style computation of maxima, rather than sums, for inside probabilities. We will refer to the probabilities of the best inside analyses computed this way as vit_β. The probability of the best parse tree is then computed as $vit\_\beta_1(1,m)$.

Alternatively, one can instantiate p and q in A1 to shorter subsequences of the input string and determine the most likely tree by a Viterbi-style computation of both inside and outside probabilities. In particular, one can instantiate p and q to a single k and dynamically compute maxima (rather than sums) for outside probabilities of a preterminal node. We will refer to the probabilities of the best outside analyses computed in this way as vit_α. The probability of the best parse tree is then computed as $vit\_\alpha_j(k,k)\ P(N^j \to w_k)$. The best tag N^j for a given word w_k inside the best parse tree can be computed by the following formula:

$$\arg\max_j vit\_\alpha_j(k,k)P(N^j \to w_k) \tag{A3}$$

Formula A3 for Viterbi parsing constitutes a specification in terms of outside probabilities that directly parallels the main components of the formula in A2 which is used in LoPar's tagging mode. Stating the relevant formulas in this way should help clarify the similarities and differences between the two LoPar modes.