A Maximum-Entropy Partial Parser for Unrestricted Text

Wojciech Skut and **Thorsten Brants**
Universität des Saarlandes
Computational Linguistics
D-66041 Saarbrücken, Germany
{skut,brants}@coli.uni-sb.de

In Proceedings of the Sixth Workshop on Very Large Corpora, Montréal, Québec, 1998.

arXiv:cmp-lg/9807006 v1 17 Jul 1998

Abstract

This paper describes a partial parser that assigns syntactic structures to sequences of part-of-speech tags. The program uses the *maximum entropy* parameter estimation method, which allows a flexible combination of different knowledge sources: the hierarchical structure, parts of speech and phrasal categories. In effect, the parser goes beyond simple bracketing and recognises even fairly complex structures. We give accuracy figures for different applications of the parser.

1 Introduction

The maximum entropy framework has proved to be a powerful modelling tool in many areas of natural language processing. Its applications range from sentence boundary disambiguation (Reynar and Ratnaparkhi, 1997) to part-of-speech tagging (Ratnaparkhi, 1996), parsing (Ratnaparkhi, 1997) and machine translation (Berger et al., 1996).

In the present paper, we describe a *partial parser* based on the maximum entropy modelling method. After a synopsis of the maximum entropy framework in section 2, we present the motivation for our approach and the techniques it exploits (sections 3 and 4). Applications and results are the subject of the sections 5 and 6.

2 Maximum Entropy Modelling

The expressiveness and modelling power of the maximum entropy approach arise from its ability to combine information coming from different knowledge sources. Given a set X of possible histories and a set Y of futures, we can characterise events from the joint event space X, Y by defining a number of *features*, i.e., equivalence relations over $X \times Y$. By defining these features, we express our insights about information relevant to modelling.

In such a formalisation, the maximum entropy technique consists in finding a model that (a) fits the empirical expectations of the predefined features, and (b) does not assume anything specific about events that are not subject to constraints imposed by the features. In other words, we search for the maximum entropy probability distribution p^*:

$$p^* = \operatorname*{argmax}_{p \in P} H(p)$$

where $P = \{p{:}p \text{ meets the empirical feature expectations}\}$ and $H(p)$ denotes the entropy of p.

For parameter estimation, we can use the Improved Iterative Scaling (IIS) algorithm (Berger et al., 1996), which assumes p to have the form:

$$p(x, y) = \frac{1}{Z(x)} \cdot e^{\sum_i \lambda_i \cdot f_i(x,y)}$$

where $f_i : X \times Y \rightarrow \{0, 1\}$ is the indicator function of the i-th feature, λ_i the weight assigned to this feature, and $Z(x)$ a normalisation constant. IIS iteratively adjusts the weights (λ_i) of the features; the model converges to the maximum entropy distribution.

One of the most attractive properties of the maximum entropy approach is its ability to cope with feature decomposition and overlapping features. In the following sections, we will show how these advantages can be exploited for *partial parsing*, i.e., the recognition of syntactic structures of limited depth.

3 Context Information for Parsing

An interesting feature of many partial parsers is that they recognise phrase boundaries mainly on the basis of cues provided by strictly local contexts. Regardless of whether or not abstractions such as phrases occur in the model, most of the relevant information is contained directly in the sequence of words and part-of-speech tags to be processed.

An archetypal representative of this approach is the method described by Church (1988), who used corpus frequencies to determine the boundaries of simple non-recursive NPs. For each pair of part-of-speech tags t_i, t_j, the probability of an NP boundary ('[' or ']') occurring between t_i and t_j is computed. On the basis of these context probabilities, the program inserts the symbols '[' and ']' into sequences of part-of-speech tags.

Information about lexical contexts also significantly improves the performance of deep parsers. For instance, Joshi and Srinivas (1994) encode partial structures in the Tree Adjoining Grammar framework and use tagging techniques to restrict a potentially very large amount of alternative structures. Here, the context incorporates information about both the terminal yield and the syntactic structure built so far.

Local configurations of words and parts of speech are a particularly important knowledge source for lexicalised grammars. In the Link Grammar framework (Lafferty et al., 1992; Della Pietra et al., 1994), strictly local contexts are naturally combined with long-distance information coming from *long-range trigrams*.

Since modelling syntactic context is a very knowledge-intensive problem, the maximum entropy framework seems to be a particularly appropriate approach. Ratnaparkhi (1997) introduces several *contextual predicates* which provide rich information about the syntactic context of nodes in a tree (basically, the structure and category of nodes dominated by or dominating the current phrase). These predicates are used to guide the actions of a parser.

The use of a rich set of contextual features is also the basic idea of the approach taken by Hermjakob and Mooney (1997), who employ predicates capturing syntactic and semantic context in their parsing and machine translation system.

4 A Partial Parser for German

The basic idea underlying our approach to partial parsing can be characterised as follows:

- An appropriate encoding format makes it possible to express all relevant lexical, categorial and structural information in a finite alphabet of *structural tags* assigned to words (section 4.1).
- Given a sequence of words tagged with part-of-speech labels, a Markov model is used to determine the most probable sequence of structural tags (section 4.2).
- Parameter estimation is based on the maximum entropy technique, which takes full advantage of the multi-dimensional character of the *structural tags* (section 4.3).

The details of the method employed are explained in the remainder of this section.

4.1 Relevant Contextual Information

Three pieces of information associated with a word w_i are considered relevant to the parser:

- the part-of-speech tag t_i assigned to w_i
- the structural relation r_i between w_i and its predecessor w_{i-1}
- the syntactic category c_i of $parent(w_i)$

On the basis of these three dimensions, *structural tags* are defined as triples of the form $S_i = \langle t_i, r_i, c_i \rangle$. For better readability, we will sometimes use attribute-value matrices to denote such tags.

$$S_i = \begin{bmatrix} \text{TAG } t_i \\ \text{REL } r_i \\ \text{CAT } c_i \end{bmatrix}$$

Since we consider structures of limited depth, only seven values of the REL attribute are distinguished.

$$r_i = \begin{cases} 0 & \text{if} \quad parent(w_i) = parent(w_{i-1}) \\ + & \text{if} \quad parent(w_i) = parent^2(w_{i-1}) \\ ++ & \text{if} \quad parent(w_i) = parent^3(w_{i-1}) \\ - & \text{if} \quad parent^2(w_i) = parent(w_{i-1}) \\ -- & \text{if} \quad parent^3(w_i) = parent(w_{i-1}) \\ = & \text{if} \quad parent^2(w_i) = parent^2(w_{i-1}) \\ 1 & \text{else} \end{cases}$$

If more than one of the conditions above are met, the first of the corresponding tags in the list is assigned. Figure 1 exemplifies the encoding format.

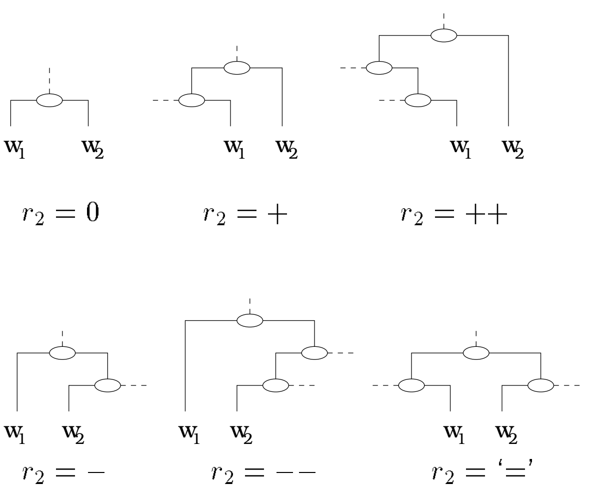

Figure 1: Tags r_2 assigned to word w_2

These seven values of the r_i attribute are mostly sufficient to represent the structure of even fairly complex NPs, PPs and APs, involving PP and genitive NP attachment as well as complex prenominal modifiers. The only NP components that are not treated here are relative clauses and infinitival complements. A German prepositional phrase and its encoding are shown in figure 2.

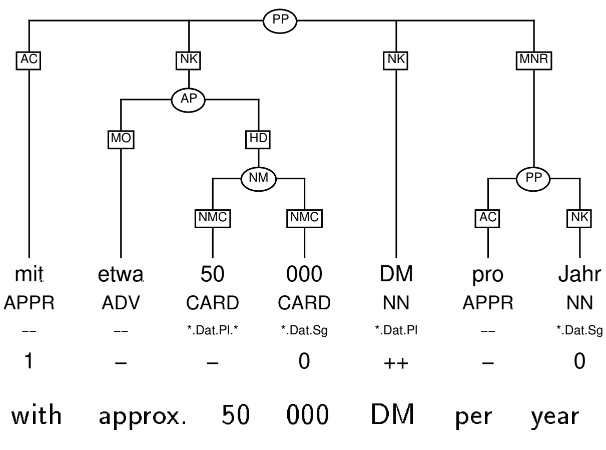

Figure 2: A sample structure. The labels are explained in Appendix B.

4.2 A Markovian Parser

The task of the parser is to determine the best sequence of triples $\langle t_i, r_i, c_i \rangle$ for a given sequence of part-of-speech tags $\langle t_0, t_1, ...t_n \rangle$. Since the attributes TAG, REL and CAT can take only a finite number of values, the number of such triples will also be finite, and they can be used to construct a 2-nd order Markov model. The triples $S_i = \langle t_i, r_i, c_i \rangle$ are states of the model, which emits POS tags (t_j) as signals.

In this respect, our approach does not much differ from standard part-of-speech tagging techniques. We simply assign the most probable sequence of structural tags $S = \langle S_0, S_1, ..., S_n \rangle$ to a sequence of part-of-speech tags $T = \langle t_0, t_1, ..., t_n \rangle$. Assuming the Markov property, we obtain:

$$\operatorname*{argmax}_S P(S|T) \tag{1}$$

$$= \operatorname*{argmax}_R P(S) \cdot P(T|S)$$

$$= \operatorname*{argmax}_R \prod_{i=1}^{k} P(S_i|S_{i-2}, S_{i-1}) P(t_i|S_i)$$

The part-of-speech tags are encoded in the structural tag (the t_i dimension), so S uniquely determines T. Therefore, we have $P(t_i|S_i) = 1$ if $S_i = \langle t_i, r_i, c_i \rangle$ and 0 otherwise, which simplifies calculations.

4.3 Parameter Estimation

The more interesting aspect of our parser is the estimation of contextual probabilities, i.e., calculating the probability of a structural tag S_i (the "future") conditional on its immediate predecessors S_{i-1} and S_{i-2} (the "history").

history		future
$\begin{bmatrix} \text{CAT: } c_{i-2} \\ \text{REL: } r_{i-2} \\ \text{TAG: } t_{i-2} \end{bmatrix}$	$\begin{bmatrix} \text{CAT: } c_{i-1} \\ \text{REL: } r_{i-1} \\ \text{TAG: } t_{i-1} \end{bmatrix}$	$\begin{bmatrix} \text{CAT: } c_{i} \\ \text{REL: } r_{i} \\ \text{TAG: } t_{i} \end{bmatrix}$

In the following two subsections, we contrast the traditional HMM estimation method and the maximum entropy approach.

4.3.1 Linear Interpolation

One possible way of parameter estimation is to use standard HMM techniques while treating the triples $S_i = \langle t_i, c_i, r_i \rangle$ as atoms. Trigram probabilities are estimated from an annotated corpus by using relative frequencies r:

$$r(S_i|S_{i-2}, S_{i-1}) = \frac{f(S_{i-2}, S_{i-1}, S_i)}{f(S_{i-2}, S_{i-1})}$$

A standard method of handling sparse data is to use a linear combination of unigrams, bigrams, and trigrams $\hat{p}$:

$$\begin{aligned}\hat{p}(S_i|S_{i-2}, S_{i-1}) &= \lambda_1 r(S_i) \\ &\quad +\lambda_2 r(S_i|S_{i-1}) \\ &\quad +\lambda_3 r(S_i|S_{i-2}, S_{i-1})\end{aligned}$$

The λ_i denote weights for different context sizes and sum up to 1. They are commonly estimated by deleted interpolation (Brown et al., 1992).

4.3.2 Features

A disadvantage of the traditional method is that it considers only full n-grams $S_{i-n+1}, ..., S_i$ and ignores a lot of contextual information, such as regular behaviour of the single attributes TAG, REL and CAT. The maximum entropy approach offers an attractive alternative in this respect since we are now free to define features accessing different constellations of the attributes. For instance, we can abstract over one or more dimensions, like in the context description in figure 1.

history		future
$\begin{bmatrix}\text{TAG: } ART\end{bmatrix}$	$\begin{bmatrix}\text{TAG: } ADJA \\ \text{REL: } 0\end{bmatrix}$	$\begin{bmatrix}\text{CAT: } NP \\ \text{REL: } 0 \\ \text{TAG: } NN\end{bmatrix}$

Table 1: A partial trigram feature

Such "partial n-grams" permit a better exploitation of information coming from contexts observed in the training data. We say that a feature f_k defined by the triple $\langle M_{i-2}, M_{i-1}, M_i \rangle$ of attribute-value matrices is *active* on a trigram context $\langle S'_{i-2}, S'_{i-1}, S'_i \rangle$ (i.e., $f_k(S'_{i-2}, S'_{i-1}, S'_i) = 1$) iff M_j unifies with the attribute-value matrix M'_j encoding the information contained in S'_i for $j = i-2, i-1, i$. A novel context would on average activate more features than in the standard HMM approach, which treats the $\langle t_i, r_i, c_i \rangle$ triples as atoms.

The actual features are extracted from the training corpus in the following way: we first define a number of *feature patterns* that say which attributes of a trigram context are relevant. All feature pattern instantiations that occur in the training corpus are stored; this procedure yields several thousands of features for each pattern.

After computing the weights λ_i of the features occurring in the training sample, we can calculate the contextual probability of a multi-dimensional structural tag S_i following the two tags S_{i-2} and S_{i-1}:

$$p(S_i|S_{i-2}, S_{i-1}) = \frac{1}{Z(x)} \cdot e^{\sum_i \lambda_i \cdot f_i(S_{i-2}, S_{i-1}, S_i)}$$

We achieved the best results with 22 empirically determined feature patterns comprising full and partial n-grams, $n \leq 3$. These patterns are listed in Appendix A.

5 Applications

Below, we discuss two applications of our maximum entropy parser: treebank annotation and chunk parsing of unrestricted text. For precise results, see section 6.

5.1 Treebank Annotation

The partial parser described here is used for corpus annotation in a treebank project, cf. (Skut et al., 1997). The annotation process is more interactive than in the Penn Treebank approach (Marcus et al., 1994), where a sentence is first preprocessed by a partial parser and then edited by a human annotator. In our method, manual and automatic annotation steps are closely interleaved. Figure 3 exemplifies the human-computer interaction during annotation.

The annotations encode four kinds of linguistic information: 1) parts of speech and inflection, 2) structure, 3) phrasal categories (node labels), 4) grammatical functions (edge labels).

Part-of-speech tags are assigned in a preprocessing step. The automatic instantiation of labels is integrated into the assignment of structures. The annotator marks the words and phrases to be grouped into a new substructure, and the node and edge labels are inserted by the program, cf. (Brants et al., 1997).

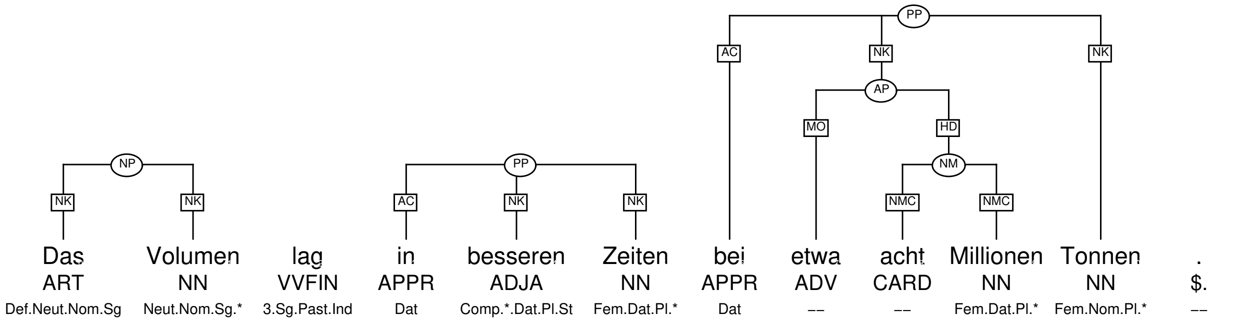

Figure 3: A chunked sentence (*in better times, the volume was around eight million tons*). Grammatical function labels: NK nominal kernel component, AC adposition, NMC number component, MO modifier.

Initially, such annotation increments were just local trees of depth one. In this mode, the annotation of the PP *bei etwa acht Millionen Tonnen* (*[at] around eight million tons*) involves three annotation steps (first the number phrase *acht Millionen*, then the AP, and the PP). Each time, the annotator highlights the immediate constituents of the phrase being constructed.

The use of the partial parser described in this paper makes it possible to construct the whole PP in only one step: The annotator marks the words dominated by the PP node, and the internal structure of the new phrase is assigned automatically. This significantly reduces the amount of manual annotation work. The method yields reliable results in the case of phrases that exhibit a fairly rigid internal structure. More than 88% of all NPs, PPs and APs are assigned the correct structure, including PP attachment and complex prenominal modifiers.

Further examples of structures recognised by the parser are shown in figure 4. A more detailed description of the annotation mode can be found in (Brants and Skut, 1998).

5.2 NP Chunker

Apart from treebank annotation, our partial parser can be used to chunk part-of-speech tagged text into major phrases. Unlike in the previous application, the tool now has to determine not only the internal structure, but also the external boundaries of phrases. This makes the task more difficult; especially for determining PP attachment.

However, if we restrict the coverage of the parser to the prenominal part of the NP/PP, it performs quite well, correctly assigning almost 95% of all structural tags, which corresponds to a bracketing precision of ca. 87%.

6 Results

In this section, we report the results of a cross-validation of the parser carried out on the NeGra Treebank (Skut et al., 1997). The corpus was converted into structural tags and partitioned into a training and a testing part (90% and 10%, respectively). We repeated this procedure ten times with different partitionings; the results of these test runs were averaged.

The weights of the features used by the maximum entropy parser were determined with the help of the Maximum Entropy Modelling Toolkit, cf. (Ristad, 1996). The number of features reached 120,000 for the full training corpus (12,000 sentences). Interestingly, tagging accuracy decreased after after 4–5 iterations of Improved Iterative Scaling, so only 3 iterations were carried out in each of the test runs.

The accuracy measures employed are explained as follows.

tags: the percentage of structural tags with the correct value r_i of the REL attribute,

bracketing: the percentage of correctly recognised nodes,

labelled bracketing: like bracketing, but including the syntactic category of the nodes,

structural match: the percentage of correctly recognised tree structures (top-level chunks only, labelling is ignored).

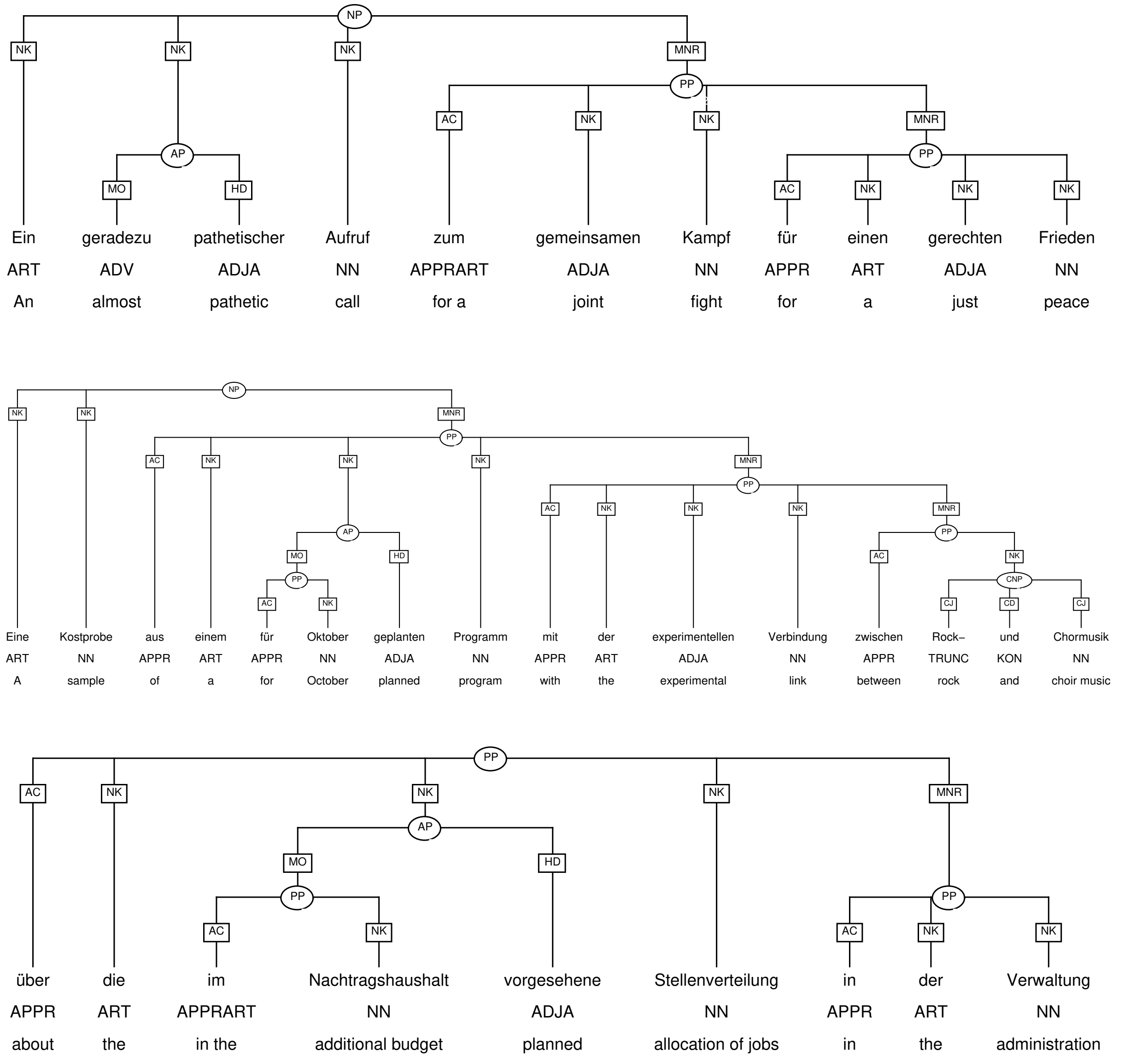

Figure 4: Examples of complex NPs and PPs correctly recognised by the parser. In the treebank application, such phrases are part of larger structures. The external boundaries (the first and the last word of the examples) are highlighted by an annotator, the parser recognises the internal boundaries and assigns labels.

6.1 Treebank Application

In the treebank application, information about the external boundaries of a phrase is supplied by an annotator. To imitate this situation, we extracted from the NeGra corpus all sequences of part-of-speech tags spanned by NPs PPs, APs and complex adverbials. Other tags were left out since they do not appear in chunks recognised by the parser. Thus, the sentence shown in figure 3 contributed three substrings to the chunk corpus: ART NN, APPR ADJA NN and APPR ADV CARD NN NN, which would also be typical annotator input. A designated separator character was used to mark chunk boundaries.

Table 2 shows the performance of the parser on the chunk corpus.

Table 2: Recall and precision results for the interactive annotation mode.

measure	total	correct	recall	prec.
tags	129822	123435	95.1%	
bracketing	56715	49715	87.7%	89.1%
lab. brack.	56715	47415	83.6%	84.8%
struct. match	37942	33450	88.2%	88.0%

6.2 Chunking Application

Table 3 shows precision and recall for the chunking application, i.e., the recognition of kernel NPs and PPs in part-of-speech tagged text. Post-nominal PP attachment is ignored. Unlike in the treebank application, there is no pre-editing by a human expert. The absolute numbers differ from those in table 2 because certain structures are ignored. The total number of structural tags is higher since we now parse whole sentences rather then separate chunks.

In addition to the four accuracy measures defined above, we also give the percentage of chunks with correctly recognised external boundaries (irrespective of whether or not there are errors concerning their internal structure).

Table 3: Recall and precision for the chunking application. The parser recognises only the prenominal part of the NP/PP (without focus adverbs such as *also*, *only*, etc.).

measure	total	correct	recall	prec.
tags	166995	158541	94.9%	
bracketing	51912	45241	87.2%	86.9%
lab. brack.	51912	43813	84.4%	84.2%
struct. match	46599	41422	88.9%	87.6%
ext. bounds	46599	43833	94.1%	93.4%

6.3 Comparison to a Standard Tagger

In the following, we compare the performance of the maximum-entropy parser with the precision of a standard HMM-based approach trained on the same data, but using only the frequencies of complete trigrams, bigrams and unigrams, whose probabilities are smoothed by linear interpolation, as described in section 4.3.1.

Figure 5 shows the percentage of correctly assigned values r_i of the REL attribute depending on the size of the training corpus. Generally, the maximum entropy approach outperforms the linear extrapolation technique by about 0.5% – 1.5%, which corresponds to a 1% – 3% difference in structural match. The difference decreases as the size of the training sample grows. For the full corpus consisting of 12,000 sentences, the linear interpolation tagger is still inferior to the maximum entropy one, but the difference in precision becomes insignificant (0.2%). Thus, the maximum entropy technique seems to particularly advantageous in the case of sparse data.

7 Conclusion

We have demonstrated a partial parser capable of recognising simple and complex NPs, PPs and APs in unrestricted German text. The maximum entropy parameter estimation method allows us to optimally use the context information contained in the training sample. On the other hand, the parser can still be viewed as a Markov model, which guarantees high efficiency (processing in linear time). The program can be trained even with a relatively small amount of treebank data; then it can be used for parsing unrestricted pre-tagged text.

As far as coverage is concerned, our parser can handle recursive structures, which is an advantage compared to simpler techniques such as that described by Church (1988). On the other hand, the Markov assumption underlying our approach means that only strictly local dependencies are recognised. For full parsing, one would probably need non-local contextual information, such as the *long-range trigrams* in Link Grammar (Della Pietra et al., 1994).

Our future research will focus on exploiting morphological and lexical knowledge for partial parsing. Lexical context is particularly relevant for the recognition of genitive NP and PP attachment, as well as complex proper names. We hope that our approach will benefit from related work on this subject, cf. (Ratnaparkhi et al., 1994). Further precision gain can also be achieved by enriching the structural context, e.g. with information about the category of the grandparent node.

8 Acknowledgements

This work is part of the DFG Collaborative Research Programme 378 *Resource-Adaptive Cog-*

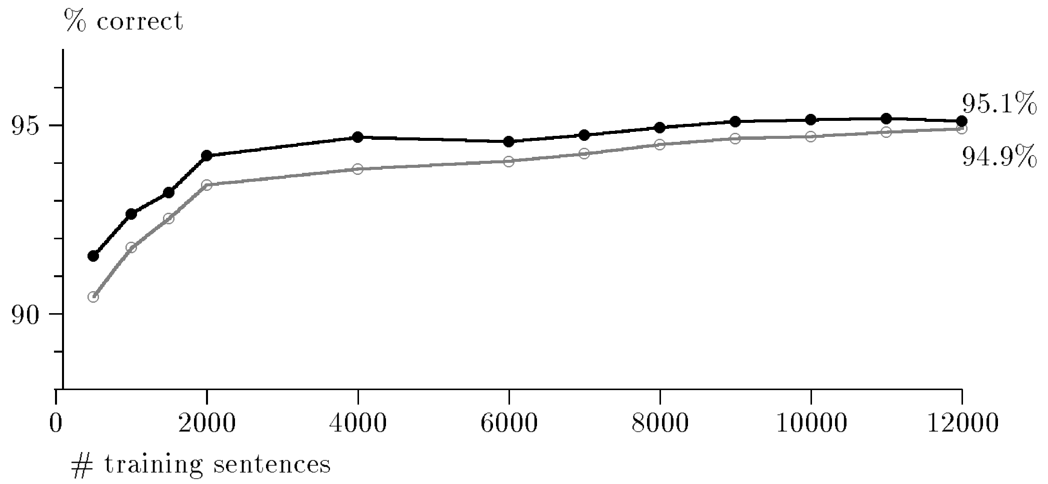

Figure 5: Tagging precision achieved by the maximum entropy parser (—●—) and a tagger using linear interpolation (—○—). Precision is shown for different numbers of training sentences.

nitive Processes, Project C3 *Concurrent Grammar Processing.*

Many thanks go to Eric S. Ristad. We used his freely available Maximum Entropy Modelling Toolkit to estimate context probabilities.

References

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics Vol. 22 No. 1*, 22(1):39–71.

Thorsten Brants and Wojciech Skut. 1998. Automation of treebank annotation. In *Proceedings of NeMLaP-3*, Sydney, Australia.

Thorsten Brants, Wojciech Skut, and Brigitte Krenn. 1997. Tagging grammatical functions. In *Proceedings of EMNLP-97*, Providence, RI, USA.

P. F. Brown, V. J. Della Pietra, Peter V. de Souza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Kenneth Ward Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA.

S. Della Pietra, V. Della Pietra, J. Gillett, J. Lafferty, H. Printz, and L. Ures. 1994. Inference and estimation of a long-range trigram model. In *Proceedings of the Second International Colloquium on Grammatical Inference and Applications, Lecture Notes in Artificial Intelligence.* Springer Verlag.

Ulf Hermjakob and Raymond J. Mooney. 1997. Learning parse and translation decisions from examples with rich contexts. In *Proceedings of ACL-97*, pages 482 – 489, Madrid, Spain.

Aravind K. Joshi and B. Srinivas. 1994. Disambiguation of super parts of speech (or supertags). In *Proceedings COLING 94*, Kyoto, Japan.

John Lafferty, Daniel Sleator, and Davy Temperley. 1992. Grammatical trigrams: A probabilistic model of link grammar. In *Proceedings of the AAAI Conference on Probabilistic Approaches to Natural Language.*

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of English: the Penn Treebank. In Susan Armstrong, editor, *Using Large Corpora.* MIT Press.

Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 250–255.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP-96*, Philadelphia, Pa., USA.

Adwait Ratnaparkhi. 1997. A linear observed time statistical parser based on maximum entropy models. In *Proceedings of EMNLP-97*, Providence, RI, USA.

Jeffrey Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of ANLP-97*, Washington, DC, USA.

Eric Sven Ristad, 1996. *Maximum Entropy Modelling Toolkit, User's Manual.* Princeton University, Princeton. Available at `cmp-lg/9612005`.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of ANLP-97*, Washington, DC.

Christine Thielen and Anne Schiller. 1995. Ein kleines und erweitertes Tagset fürs Deutsche. In *Tagungsberichte des Arbeitstreffens Lexikon + Text 17./18. Februar 1994, Schloß Hohentübingen. Lexicographica Series Maior*, Tübingen. Niemeyer.

Appendix A: Feature Patterns

Below, we give the 22 n-gram feature patterns used in our experiments.

	history		future
Trigram features	r, t, c	r, t, c	r, t, c
	r^{sibl}, t, c	r, t, c	r, t, c
	r, c	r, c	r, t, c
	t	r^{sibl}, c	r, t, c
	r^{sibl}, t	r, t	r, t
	r, t	r, t, c	r, t
	r, c	r, t, c	r, c
	r	r, t, c	r
	r, t, c	r, t, c	r, t
	r, t, c	r, t, c	r, c
	t	r, t, c	r, c
Bigram features		r, t, c	r, t, c
		r, t, c	r, t
		r, t, c	r, c
		r^{sibl}, t	r, t
		r, t	r, t
		c	r, c
		t	r, t
		r	r
Unigram features			r, t, c
			r, t
			r^{sibl}, t

The symbols r (REL), t (TAG), and c (CAT) indicate which attributes are taken into account when generating a feature according to a particular pattern. r^{sibl} is a binary-valued attribute saying whether the word under consideration and its immediate predecessor are siblings (i.e., whether or not $r = 0$).

Appendix B: Tagsets

This section contains descriptions of tags used in this paper. These are *not* complete lists.

B.1 Part-of-Speech Tags

We use the Stuttgart-Tübingen-Tagset. The complete set is described in (Thielen and Schiller, 1995).

ADJA	attributive adjective
ADV	adverb
APPR	preposition
APPRART	preposition with determiner
ART	article
CARD	cardinal number
KON	Conjunction
NE	proper noun
NN	common noun
PROAV	pronominal adverb
TRUNC	first part of truncated noun
VAFIN	finite auxiliary
VAINF	infinite auxiliary
VMFIN	finite modal verb
VVFIN	finite verb
VVPP	past participle of main verb

B.2 Phrasal Categories

AP	adjective phrase
MPN	multi-word proper noun
NM	multi token numeral
NP	noun phrase
PP	prepositional phrase
S	sentence
VP	verb phrase

B.3 Grammatical Functions

AC	adpositional case marker
HD	head
MO	modifier
MNR	post-nominal modifier
NG	negation
NK	noun kernel
NMC	numerical component
OA	accusative object
OC	clausal object
PNC	proper noun component
SB	subject