Tagging Complex Non-Verbal German Chunks with Conditional Random Fields

Luzia Roth
Institute of Computational Linguistics
University of Zurich
luzia.roth@uzh.ch

Simon Clematide
Institute of Computational Linguistics
University of Zurich
simon.clematide@cl.uzh.ch

Abstract

We report on chunk tagging methods for German that recognize complex non-verbal phrases using structural chunk tags with Conditional Random Fields (CRFs). This state-of-the-art method for sequence classification achieves 93.5% accuracy on newspaper text. For the same task, a classical trigram tagger approach based on Hidden Markov Models reaches a baseline of 88.1%. CRFs allow for a clean and principled integration of linguistic knowledge such as part-of-speech tags, morphological constraints and lemmas. The structural chunk tags encode phrase structures up to a depth of 3 syntactic nodes. They include complex prenominal and postnominal modifiers that occur frequently in German noun phrases.

1 Introduction

In this paper[1], we report on comprehensive experimental results for a chunk tagging approach that recognizes complex non-verbal phrases such as nominal phrases (NP), prepositional phrases (PP), adjectival and adverbial phrases in German. We go beyond simple base chunks, that is, non-recursive and non-overlapping sequences of words. Base chunks were introduced and formalized as a sequence classification problem by Ramshaw and Marcus (1995) and popularized by a CoNLL shared task on chunking (Tjong Kim Sang and Buchholz, 2000). This problem is also known as the `IOB` chunk tagging problem because the chunk layer can be formulated as a sequence of tags expressing the begin (`B`) and continuation (`I`) of a chunk, or whether a token is viewed as being outside (`O`) of any chunk.

In contrast to the base chunk approach, we analyze the internal structure of complex phrases up to a maximal depth of 3 phrase structure nodes. As introduced by Skut and Brants (1998), structural chunk tags are needed that encode the hierarchical relation between adjacent tokens. Both, the `IOB` and the structural chunk tag approach can be treated as a sequence classification problem. We compare the performance of well-established sequence classifiers such as Hidden Markov Models (HMMs) with the state-of-the-art method of Conditional Random Fields (CRFs) on the TüBa-D/Z treebank (Telljohann et al., 2004), which is the largest collection of consistently annotated newspaper sentences in German.

The paper is organized as follows: In Section 2, we introduce the idea of structural chunk tags and present the data extraction and transformation from the treebank as well as the automatic linguistic enrichment of the raw data in preparation to the experiments. In addition, we describe the statistical tools and models used in our cross-validation experiments. In Section 3, we report the quantitative results of the experiments and discuss qualitative aspects of the most frequent errors.

2 Methods

Our approach is based on early work of Skut and Brants (1998). They introduced the term *chunk*

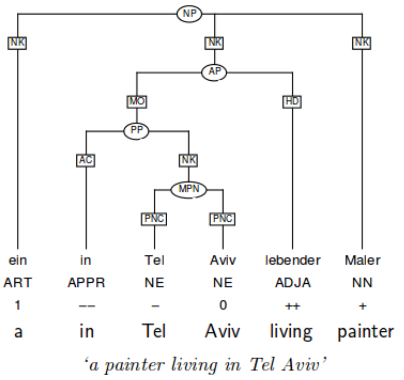

Figure 1: Complex NP annotated with structural tags as presented in Skut and Brants (1998). See Section 2.2.2 for an explanation of their chunk tags.

tagging for applying standard statistical PoS tagging techniques (i.e. HMMs) to the problem of chunking complex NPs and PPs. We extend their approach by using more data, more linguistic features, and more advanced statistical sequence classification methods to deal with this problem. Additionally, we investigate the question of how well post-nominal PPs can be identified by our improved approach.

2.1 Related Work

Skut and Brants (1998) developed a recognizer for complex chunk structures in order to create a tool for semi-automatic syntactic annotation. Their main idea was to extend chunking from a simple recognition of the boundaries of flat chunks to the calculation of nested chunk-internal syntactical structures. Given the outer boundaries of a chunk by a human annotator, their annotation system built the internal structures of chunks as complex as the one shown in Figure 1. They also evaluated their chunk tagger as a stand-alone application without human indication of chunk boundaries. This is more comparable to our experimental setting. They reached 90.9% of correctly tagged tokens using the NEGRA treebank (Skut et al., 1997) with a training corpus of 12,000 sentences. Due to the difficulties introduced by post-nominal attachment of NPs, PPs and focus adverbs, they trained and evaluated a chunk tagger without attachment of post-nominal NPs, PPs and adverbs. For this less complex task, they report a precision of 95.5%.

It is noteworthy that structural chunk tags can handle complex prenominal constructions as shown in Figure 1. IOB-style chunks typically need to disconnect the indefinite article from the nominal head of the NP (see Kübler et al. (2010) for a workaround). The NEGRA-derived German chunk tagger for flat noun, prepositional and verb chunks built on top of the TreeTagger (Schmid, 1994) shows exactly these limitations.

The recursive chunker from Kermes and Evert (2002) is based on a symbolic regular expression grammar and handles even complex prenominal constructions. It also deals with post-nominal NP attachment, but excludes post-nominal PP attachment due to the high degree of ambiguity.

Chunkers based on cascaded rules (e.g. Müller (2007)) or finite state transducer (for a more recent implementation see Barbaresi (2013)) can efficiently build shallow syntactic structure. Hinrichs (2005) contains an overview of several earlier approaches for German.

2.2 Data

For our experiments, we use the TüBa-D/Z corpus version 7.0, containing 65,524 sentences (henceforth referred to as TüBa)[2]. The corpus consists of newspaper articles with detailed morphological and syntactic annotations. This treebank is the largest for German and because of its topological and context-free grammar there are no discontiguous phrase structures as for example in the TIGER treebank (Brants et al., 2004).

2.2.1 Data Transformation and Enrichment

As can be seen in the upper tree of Figure 2, TüBa's phrase structures are deeply nested. For instance, the proper name 'Taake' is embedded at a depth of 6 phrase structure nodes. In order to be able to treat such complex PPs with our approach of limited chunk depth, we need to flatten the TüBa trees in the style of TIGER trees. The following transformations were applied:

1. The constituents of the dependent NP of a preposition are treated as immediate constituents of the PP. This approach has also been followed recently in the setting of multilingual dependency treebanks (McDonald et al., 2013).

[2]http://sfs.uni-tuebingen.de/ascl/ressourcen/corpora/tueba-dz.html

Tagset from Skut and Brants		*Internal tagset (preceding / succeeding token): p/s*			
		External tagset: p		*External tagset: s*	
0	if $m(w_i) = m(w_{i-1})$	e	if $m(w_i) = m(w_{i-1})$	e	if $m(w_i) = m(w_{i+1})$
+	if $m(w_i) = m^2(w_{i-1})$	r	if $m(w_i) = m^2(w_{i-1})$	r	if $m(w_i) = m^2(w_{i+1})$
++	if $m(w_i) = m^3(w_{i-1})$	R	if $m(w_i) = m^3(w_{i-1})$	R	if $m(w_i) = m^3(w_{i+1})$
-	if $m^2(w_i) = m(w_{i-1})$	l	if $m^2(w_i) = m(w_{i-1})$	l	if $m^2(w_i) = m(w_{i+1})$
--	if $m^3(w_i) = m(w_{i-1})$	L	if $m^3(w_i) = m(w_{i-1})$	L	if $m^3(w_i) = m(w_{i+1})$
=	if $m^2(w_i) = m^2(w_{i-1})$	E	if $m^2(w_i) = m^2(w_{i-1})$	E	if $m^2(w_i) = m^2(w_{i+1})$
1	else	-	not integrated into syntax structure	-	not integrated into syntax structure
		0	removed from syntax structure	0	removed from syntax structure
		x	chunk boundary	x	chunk boundary

Table 1: Comparison between Skut and Brants' tagset and our tagsets. Our data contains 50 different p/s tags out of 81 possible combinations.

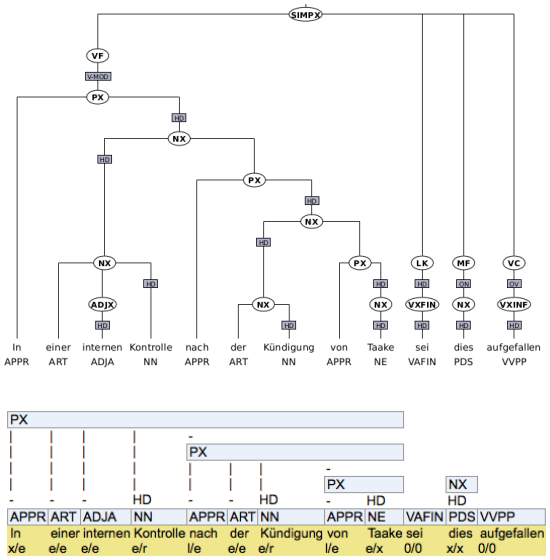

Figure 2: Example for the transformation of a deep syntactic phrase structure to the flattened chunk format. The structural chunk labels with our internal chunk tagset are on the last line. The shown sentence fragment translates as "*In an internal control after the termination of Taake this was noticed,...*".

2. The content of unary nodes which are non-heads in their mother constituent is directly attached to the mother node.

3. Coordinated unary nodes are directly attached to their mother nodes.

After the application of these transformations, all topological and verbal constituents were removed from the syntactic trees. All remaining phrase structures with a syntactic depth larger than 3 were removed. The final result of these transformations for the example sentence is shown in the lower part of Figure 2.

2.2.2 Internal and External Chunk Tagsets

For our experiments, we work with an enriched internal chunk tagset that encodes the structural relation of a token to its preceding (p) and succeeding (s) token. More fine-grained internal tagsets have proved to be profitable for statistical tagging approaches in the past (Brants, 1997). One goal of our experiments was to check whether this is also the case for chunk tags.

Table 1 compares Skut and Brants' tagset and our tagsets. An equation as $m(w_i) = m^2(w_{i-1})$ reads as 'the mother node of token w at position i is the grandmother node of the preceding token'. The depth of the hierarchical dominance relation m is given by its superscript. i specifies the linear position of a word in a sentence. Punctuation is never integrated in the syntactic structure (marked as '-'). Tokens connected to nodes (e.g. verbal) that were removed from the syntax structure are marked as '0'. Chunk tag 'x' indicates chunk boundaries. Figure 2 shows an example of the chunk encoding.

In our bidirectional internal tagset, an error often affects two tokens. This deteriorates the evaluation results because a single error will be counted twice. In a sentence like 'Aber weil der Koffer in einem unterirdischen See gelandet ist, [...]' ('*But because the suitcase has landed in an underground lake, [...]*') our system attaches 'in einem unterirdischen See' erroneously as a postnominal PP, resulting in two errors in the internal tagset as shown in Table 2. However, reducing the internal tagset to one of the external tagsets does not lead to a loss of information for the chunk structure. Therefore, we can train and label on the bidirectional internal tagset and map to an external before evaluation.

Tokens	*Gold*	*System*	*p/s*	*p/*	*/s*
der	x/e	x/e			
Koffer	e/x	e/r	X		X
in	x/e	l/e	X	X	
einem	e/e	e/e			
unterirdischen	e/e	e/e			
See	e/x	e/x			

Table 2: Error propagation in the internal tagset

Code	*Description*
K/k	With case/Without case
N/n	Nominative admissible/excluded
A/a	Accusative admissible/excluded
D/d	Dative admissible/excluded
G/g	Genitive admissible/excluded
U/u	With number/Without number
S/s	Singular admissible/excluded
P/p	Plural admissible/excluded

Table 3: Encoding of morphological constraints

Token	*PoS*	*Lemma*	*Morphology*	*Chk*
In	APPR	in	KnADgusp	x/e
einer	ART	ein	KnaDGUSp	e/e
internen	ADJA	intern	KNADGUSP	e/e
Kontrolle	NN	Kontrolle	KNADGUSp	e/r
nach	APPR	nach	KnaDgusp	l/e
der	ART	d	KNaDGUSP	e/e
Kündigung	NN	Kündigung	KNADGUSp	e/r
von	APPR	von	KnaDgusp	l/e
Taake	NN	Taake	????????	e/x
sei	VAFIN	sein	knadgUSp	0/0
dies	PDS	dies	KNAdgUSp	x/x
aufgefallen	VVPP	auffallen	knadgusp	0/0
,	$,	,	$$$$$$$$	-/-

Table 4: Representation of linguistic evidence and outcome (= column 'Chk')

2.2.3 Input Data Enrichment

The task of a chunk tagger is to compute the sequence of chunk tags (=outcome) for a given sequence of tokens (=evidence). However, directly using the raw text as the only evidence for predicting the outcome misses useful linguistic generalizations that are beneficial for this task. Therefore, we automatically enrich the raw text by PoS tags, normalized lemmas and morphological constraints.

First, we apply the TreeTagger 3.2 (Schmid, 1994) to compute PoS tags and lemmas from the raw text input. For unknown words, we use the tokens as the lemma. In order to reduce the sparse data problem, all lemmas are further normalized by reducing hyphenated compounds to their last segment, for instance '0:2-Niederlage' (*0-2 defeat*) is normalized to 'Niederlage'.

Second, morphological constraints for each PoS-tagged token are built from the output of GERTWOL, a commercial morphological analyzer (Koskeniemmi and Haapalainen, 1996). Morphological information is restricted to case and number and filtered according to the PoS tag computed by the TreeTagger. Because GERTWOL and the TreeTagger have slightly different categorizations of parts of speech, some tag mapping was necessary.

In German, word forms exhibit a lot of syncretism, especially between accusative and nominative case. In our current approach, we do not attempt to guess the correct analysis out of all admissible analyses, but we strive for a compact representation of the admissible as well as the excluded morphological categories. An 8 character string is used to encode these constraints in a systematic way where upper-case letters denote the admission of a category and lower-case letters denote the exclusion. Table 3 shows the actual encoding conventions. The morphological constraints of 'Häuser' (*houses*) are 'KNAdGUsP'. Word forms not known by GERTWOL are encoded by '????????' and punctuation tokens by '$$$$$$$$'.

Table 4 shows the result of the data enrichment process for our example sentence. In our experiments, we are interested to estimate the performance increase in chunk tagging that results from the morphological information, the PoS layer and the lemmas.

2.3 Tagging Structural Chunk Tags

As mentioned before, complex chunk structures in a sentence can be expressed by chunk tag sequences that correspond to the token sequence. Therefore, any sequence classification method can be applied to this problem. In our experiments, we focus on baseline methods based on HMM techniques and on state-of-the-art methods based on CRFs.

2.3.1 Chunk Tagging with Trigram Taggers

As a baseline, we use the HMM-based trigram tagger hunpos (Halácsy et al., 2007). This tool is an open-source reimplementation of the TnT tagger (Brants, 2000) that Skut and Brants (1998) developed and used for their work (see Section 2.1). A standard PoS tagger as hunpos has a predefined

and limited model how the evidence for the classification of the outcome is used. In a typical trigram setting, this is the current token (lexical emission probability) and the preceding two outcome labels (transition probability predicted from the limited history of Markov models). These restrictions guarantee a very efficient training and labeling. Additionally, there is no need for a development set for training, which enables the user to split the available tagged material into a large training (90%) and a test set (10%). As an extension to the classical trigram tagging model, the hunpos tagger allows for condition the emission probability of a word w_i on the preceding and the current tag ($P(w_i|t_{i-1}t_i)$). This second order emission probability produced consistently better results in our chunk tagging experiments than a simple first order emission probability.

A disadvantage of HMM taggers is their restriction to a single layer of evidence. For instance, if we want to predict the chunk tags from the layer of PoS and morphology, we need to integrate that information in one combined evidence token. For example, in order to chunk the third token 'internen' from our example sentence based on the evidence of PoS and morphology we would encode the evidence layer as 'NN_KNADGUSP', i.e. the concatenation of PoS and morphology. CRFs are a lot more general in that respect, as they allow to have as many separate evidence layers as needed and to combine them freely into features.

2.3.2 Chunk Tagging with sequential CRFs

Sequential Conditional Random Fields (Sutton and McCallum, 2012) are state-of-the-art sequence classification models for typical NLP problems and have been shown to deliver excellent performance on the `IOB`-style chunking tasks (Sha and Pereira, 2003).

In our experiments, we use the freely available and efficient CRF tool wapiti (Lavergne et al., 2010). Unlike HMM tools, wapiti needs handcrafted feature templates that specify which information from which evidence layer is selected and combined in order to predict the outcome, i.e. the most probable sequence of structural tags for a sentence. Feature templates are a practical abstraction layer that allow the user to specify the model in a concise way without actually forcing the user to precompute the instantiated features for each position in the sequence. The CRF tool automatically instantiates the templates with the training material. During training, it learns the optimal weights for the instantiated features, and by using appropriate regularization, it is able to filter out irrelevant features. In all experiments reported in the evaluation section, we used the default settings of wapiti: L-BGFS for the optimization of the feature weights and elastic-net for regularization. wapiti requires a development set for training, therefore, the data was split into a training (72%), development (18%) and a test (10%) set.

Relative Position	*PoS Layer Example*
Current	NN
Preceding	ADJA
Succeeding	APPR
Preceding and current	ADJA/NN
Current and succeeding	NN/APPR
Preceding, current and succeeding	ADJA/NN/APPR
Two positions back and current	ART/NN
Current and two positions forth	NN/ART

Table 5: Local context of our best CRF feature template model. The second column illustrates the template with the instantiation on the PoS layer on position 4 (token 'Kontrolle') in our example sentence from Table 4.

Our best feature model. All evidence columns shown in Table 4 can be used to define feature templates. For a given position in the sequence, evidence from the current, preceding or succeeding positions can be combined. The amount and source of evidence packed in a feature is unbound in principle, however, for performance reasons evidence from the local context is most useful. In typical sequential CRF modeling tools, the evidence features can be automatically conditioned on outcome bigrams (preceding and current token, similar to the emission order of two of HMMs) or outcome unigrams (current token only). Bigram features can easily lead to feature explosion, long training times, and decreased performance (sparse data problem). We performed extensive tests for building an optimal set of feature templates. To our own surprise, a uniform and elegant set of unigram feature templates proved to be the best. The evidence layer of tokens could be ignored totally. For the layer of

PoS, lemmas and morphological constraints, we have exactly the same feature templates[3]. Table 5 shows the local context involved in our features and illustrates them by examples taken from the PoS layer. Only one bigram feature was used, namely the bigram output distribution of the chunk tags.

Alternative or more complex additional feature templates could not improve the performance. We tested for specific morphological cases (e.g. genitive), pattern matching for function words (e.g. articles), or combinations of evidence from PoS/morphology and lemma/morphology.

Our 25 feature templates instantiate about 118 million features (standard deviation (SD) 331,577) out of which the final model contains on average 690,540 active ones (SD 134,916). The rather high SD is due to the lemma features.

3 Results and Discussion

We present selected comparative evaluation results derived from 10-fold cross-validation experiments.[4] We give the mean tagging accuracy, standard deviation and confidence intervals (CI 95%) derived from a t-test applied to the means of the 10 test folds. The CI expresses that there is a 95% chance that the true accuracy in all representative texts is contained within the computed CI.

3.1 Quantitative Evaluation

Table 6 shows the results of our evaluation. The best system with 93.54% accuracy is our wapiti model using PoS tags, morphological constraints and lemmas evaluated on the external tagset s. We outperform the hunpos baseline based on PoS evidence (87.15%) by 7.3%. Compared to the best hunpos system (88.13%) using PoS and morphology, we get an improvement of 6.1%. As expected, HMM-based tagging cannot make use of complex input tokens that combine lemma, PoS and morphology. However, the CRF model can make use of the lemma evidence resulting in a relative improvement of 1.63% compared to PoS and morphology.

Internal and external tagsets. As mentioned in Section 2.2.2, we expect the internal tagset to have a lower accuracy than the external due to error duplication. Tagset s is consistently slightly better than tagset p (the one more related to Skut and Brants') with the one exception for wapiti using PoS evidence only. The use of an enriched internal tagset proves to be beneficial. For the best system, performance is about 0.5% higher using the internal tagset. The difference is not overwhelming but appears to be very stable across all system combinations.

Upper bound by gold PoS tags and morphology. The lower part of Table 6 shows the effect of providing the correct (gold) PoS tags and morphological information (case and number) from the TüBa as evidence for the statistical tools. Using these results we can estimate the upper bound of the performance if we improve the PoS tagging and provide a better morphological disambiguation. For wapiti and our best feature templates, this is 95.15%, resulting in a maximal relative improvement of 1.72%. For hunpos, the gold information improves by maximally 1.44% for the best evidence (P,M). These rather small numbers show that there is not much room for improvement by optimizing the linguistic enrichment because there will always remain wrong PoS tags and morphological analyses.

Learning curve of internal tagset. In order to check whether more training data could lead to better results, we performed an additional experiment on the first fold using the best wapiti system. Starting with only 10,000 sentences of the TüBa, we obtain 87.08% correctly tagged tokens. Going up to 60,000 sentences, we reach 89.05%. As shown in Figure 3, the learning curve does not yet level off and more data will probably help.

3.2 Qualitative Error Analysis

In order to better understand the error types of the best system, we randomly sampled 10 errors for each of the 7 most frequent error types (see Table 7) from the test set of the first fold. In Table 8, we give a breakdown of the linguistic properties

[3]The actual wapiti code for the feature templates can be downloaded from `http://kitt.cl.uzh.ch/kitt/chunktag/wapiti.txt`.

[4]See Vanwinckelen and Blockeel (2012) for arguments why repeated cross-validation does not lead to better model estimates than simple cross-validation.

Tagger	*Evidence*	*Internal Tagset p/s*				*External Tagset p*				*External Tagset s*			
		Acc.	*SD*	CI_l	CI_u	Acc.	*SD*	CI_l	CI_u	Acc.	*SD*	CI_l	CI_u
wapiti	P,M,L	89.08	0.41	88.79	89.37	93.47	0.32	93.24	93.70	**93.54**	0.33	93.31	93.78
						92.67	0.28	92.46	92.87	93.02	0.31	92.80	93.24
	P,M	86.60	0.39	86.32	86.88	91.92	0.30	91.70	92.13	92.04	0.29	91.84	92.25
						90.95	0.27	90.76	91.14	91.40	0.30	91.18	91.61
	P	84.62	0.45	84.30	84.94	**90.89**	0.30	90.68	91.10	90.74	0.33	90.51	90.98
						89.76	0.35	89.51	90.01	90.43	0.34	90.19	90.67
hunpos	P,M,L	79.15	0.45	78.83	79.47	86.91	0.36	86.65	87.18	87.17	0.37	86.90	87.43
						84.24	0.34	84.00	84.49	87.34	0.39	87.07	87.62
	P,M	80.40	0.50	80.05	80.75	87.89	0.37	87.63	88.16	**88.13**	0.38	87.86	88.40
						85.04	0.39	84.76	85.32	87.63	0.40	87.34	87.91
	P	78.73	0.54	78.34	79.11	86.93	0.39	86.65	87.21	87.15	0.40	86.87	87.43
						83.80	0.37	83.54	84.07	86.61	0.42	86.31	86.91
Using gold PoS (GP) and gold morphology (GM)													
wapiti	GP,GM,L	91.46	0.30	91.24	91.67	95.12	0.24	94.95	95.30	**95.15**	0.24	94.98	95.33
						94.37	0.26	94.19	94.56	94.55	0.25	94.37	94.73
	GP,GM	89.21	0.35	88.96	89.46	93.83	0.26	93.64	94.01	93.87	0.25	93.70	94.05
						92.90	0.25	92.72	93.08	93.07	0.29	92.87	93.28
hunpos	GP,GM,L	80.38	0.40	80.09	80.67	88.10	0.31	87.88	88.33	88.25	0.30	88.04	88.46
						85.73	0.34	85.49	85.98	88.30	0.33	88.06	88.53
	GP,GM	81.94	0.47	81.61	82.28	89.24	0.34	89.00	89.48	**89.40**	0.35	89.15	89.65
						86.19	0.36	85.93	86.45	88.83	0.38	88.56	89.11

Table 6: Evaluation results of 10-fold cross validation experiments. Mean accuracy, standard deviation (SD) and confidence interval 95% (CI_l, CI_u) are reported. The evidence column specifies the type of evidence used for training and testing: P=PoS, M=morphological constraints, L=lemmas. Rows without numbers for the internal tagset indicate experiments where we trained directly on the external tagsets.

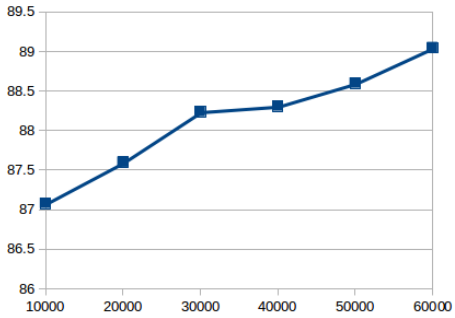

Figure 3: Learning curve of internal tagset

Wrong	*Count*	*Error Type*	*Correct*
l/e	730	Attachment	x/e
e/r	648	Attachment	e/x
x/e	626	No attachment	l/e
e/e	551	Attachment one level lower	l/e
e/x	514	No attachment	e/E
e/x	432	No attachment	e/r
x/e	406	Attachment one level lower	x/r
...	...	...	...
All	12,998		

Table 7: Most frequent error types of the internal tagset (from about 500 error types)

for the two main sources of mistakes, namely attachment errors (47 of 70) and errors in the attachment level (19 of 70). The 4 remaining cases are due to inconsistent tag sequences.

Attachment errors. In 27 cases an attachment is missing, 20 cases have wrong attachments. This error type is mostly related to PPs, followed by NPs, and adjectival phrases (APs). Furthermore, our system often has difficulties with attachment in combination with conjuncts, appositions and comparisons (see Table 8).

	Count	*PP*	*NP*	*AP*
Attachment	**20**	19	1	
thereof ambiguous	4	4		
No attachment	**27**	18	7	2
thereof with conjuncts	8	3	5	
thereof comparisons	2	1	1	
thereof with appositions	1		1	

Table 8: Attachment errors

Errors related to the level of attachment. In these cases, our system attaches a level lower than the gold standard. 12 of 19 cases are shallowly embedded prepositions, most of the time combined with conjuncts and appositions.

Figure 4 shows a case where the material of

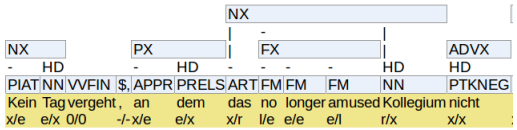

Figure 4: Sentence 330 with gold chunk tags: "*Not a day goes by that the no longer amused college does not...*"

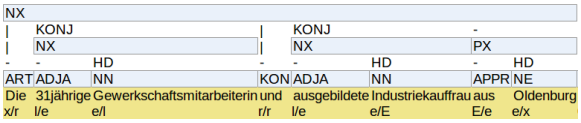

Figure 5: Sentence 78: "*The 31 year-old union employee and industrial merchant from Oldenburg . . .*"

the gold standard phrase 'FX' ('foreign language material') was directly embedded in the nominal chunk 'NX' by our system. It assigned to following chunk tags: *das*_x/e *no*_e/e *longer*_e/e *amused*_e/e *Kollegium*_e/x.

Another possibility is that appositions, conjuncts or APs are not recognized as such and the respective tokens are embedded on the same level as the rest of the chunk.

Another source of errors are conjuncts where some tokens are assigned to another conjunct than annotated in the TüBa. Figure 5 shows a case where the determiner 'Die' ('the') is integrated in the first conjunct by our system: *Die*_x/e *31jährige*_e/e *Gewerkschaftsmitarbeiterin*_e/l . . . .

Inconsistent tag sequences. As the bidirectional chunk tags encode the relation to the preceding and the succeeding token, the forward-looking tag part of one token defines the backward-looking part of the following token. Our system assigns inconsistent tag sequences in 4 cases. In all cases this involves punctuation marks inside a chunk. Figure 6 shows the gold standard where our system predicted the following inconsistent (in bold) chunk tags: *der*_x/e *Frankfurter*_e/e *"*_-/- *Guru*_e/**x** *"*_-/- *Berthold*_**E**/e *Kilian*_e/E.

4 Conclusion

With our experiments we have shown that a CRF-based state-of-the-art statistical sequence tagger as wapiti using our hand-crafted feature templates

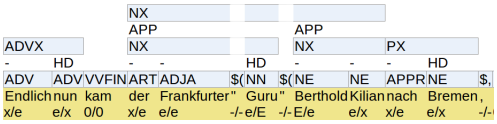

Figure 6: Sentence 211: "*Finally the "Guru" Berthold Kilian from Frankfurt came to Bremen, to...*"

can solve the structural chunk tagging problem for German with an accuracy of 93.5%. For the same task, a classical HMM-based trigram tagger reaches only 88.1% accuracy and is therefore substantially outperformed. Standard HMM tools cannot easily profit from additional evidence such as lemmas. Our results for HMMs are not directly comparable with the reported accuracy of 90.9% of Skut and Brants (1998). Their HMM system additionally includes carefully selected morphological and syntactic information.

Our final feature templates for chunk tagging turned out to be concise and uniformly structured across the evidence layers of PoS, morphological constraints and lemmas. Features and feature combinations from a local context of maximally two tokens to the left and right of the current token turned out to be optimal. Although we tried our best, there might be some unexplored optimizations. However, we would not expect substantial improvements using the same sources of evidence that we experimented with.

The learning curve for the best system suggests that more training material can improve the results even further. More training material could be provided by transforming the TIGER treebank and/or the NEGRA treebank into chunk structures comparable to the ones derived from the TüBa.

The evaluation of the internal tagset with its 50 different tags showed 504 different tag confusions for the test set of our first fold. However, the majority of the occurring errors are attachment errors and most of them are rather unsurprisingly PP attachment errors. Although more training data will probably result in some improvement, a more principled approach for the PP attachment problem seems necessary (see Van Asch and Daelemans (2009)). Within the framework of CRFs an especially crafted evidence column for verb/preposition preferences could be feasible. However, given the progress in efficient de-

pendency parsing we should carefully consider the combination of local evidence – which is typically exploited by approaches as ours – and non-local evidence which is needed for full parsing (see Swift et al. (2004)).

Our experiments with perfect morphology and PoS tags from the TüBa show that better morphological evidence can slightly improve chunk tagging. However, our morphological constraints on case and number for each token realized a lot of the theoretically achievable performance gain. A practical approach of testing the effective gain using currently available resources could be the application of the German rftagger that assigns PoS and morphological tags (Schmid and Laws, 2008).

References

Adrien Barbaresi. 2013. A one-pass valency-oriented chunker for German. In *Human Language Technologies as a Challenge for Computer Science and Linguistics, Proceedings of the 6th Language & Technology Conference, Poznan, Poland*, pages 157–161.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. Tiger: Linguistic interpretation of a german corpus. *Research on Language and Computation*, 2(4):597–620.

Thorsten Brants. 1997. Internal and external tagsets in part-of-speech tagging. In *Proceedings of Eurospeech*, pages 2787–2790.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*, pages 224–231.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. Hunpos: an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 209–212. Association for Computational Linguistics.

Erhard Hinrichs. 2005. Finite-state parsing of German. In *Inquiries into Words, Constraints, and Contexts*, pages 35–44. CSLI Publications.

Hannah Kermes and Stefan Evert. 2002. Yac - a recursive chunker for unrestricted german text. In *LREC*.

Kimmo Koskeniemmi and Mariikka Haapalainen, 1996. *GERTWOL - Lingsoft Oy*, pages 121–140. Number 34 in Sprache und Information. Niemeyer Max Verlag GmbH.

Sandra Kübler, Kathrin Beck, Erhard Hinrichs, and Heike Telljohann. 2010. Chunking german: an unsolved problem. In *Proceedings of the Fourth Linguistic Annotation Workshop*, LAW IV '10, pages 147–151, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July.

Ryan McDonald, J Nivre, Y Quirmbach-Brundage, Y Goldberg, D Das, K Ganchev, K Hall, S Petrov, H Zhang, O Täckström, C Bedini, N Bertomeu Castelló, and J Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria, August. Association for Computational Linguistics.

Frank Henrik Müller. 2007. *A finite-state approach to shallow parsing and grammatical functions annotation of German*. Ph.D. thesis.

L. A. Ramshaw and M. P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Annual Workshop on Very Large Corpora*, pages 82–94. ACL.

Helmut Schmid and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 777–784, Manchester, UK, August.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 134–141.

Wojciech Skut and Thorsten Brants. 1998. Chunk tagger – statistical recognition of noun phrases. In *Proceedings of the ESSLLI Workshop on Automated Acquisition of Syntax and Parsing*, Saarbrücken, Germany.

Wojech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*, pages 88–95, Washington, D.C.

Charles A. Sutton and Andrew McCallum. 2012. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373.

Mary Swift, James Allen, and Daniel Gildea. 2004. Skeletons in the parser : Using a shallow parser to improve deep parsing. In *COLING'04*.

H. Telljohann, E. Hinrichs, S. Kübler, et al. 2004. The TüBa-D/Z treebank: Annotating German with a context-free backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2229–2235.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132. Lisbon, Portugal.

Vincent Van Asch and Walter Daelemans. 2009. Prepositional phrase attachment in shallow parsing. In *Proceedings of the International Conference RANLP-2009*, pages 12–17, Borovets, Bulgaria, September.

Gitte Vanwinckelen and Hendrik Blockeel. 2012. On estimating model accuracy with repeated cross-validation. In *BeneLearn 2012: Proceedings of the 21st Belgian-Dutch Conference on Machine Learning, Ghent, 24-25 May 2012*, pages 39–44.