Published in: First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Language (SPMRL-SANCL 2014). August 2014. Dublin, Ireland. Pp. 1-14.

Parsing German: How Much Morphology Do We Need?

Wolfgang Maier
Heinrich-Heine-Universität Düsseldorf
Düsseldorf, Germany
maierw@hhu.de

Sandra Kübler
Indiana University
Bloomington, IN, USA
skuebler@indiana.edu

Daniel Dakota
Indiana University
Bloomington, IN, USA
ddakota@indiana.edu

Daniel Whyatt
Indiana University
Bloomington, IN, USA
dwhyatt@indiana.edu

Abstract

We investigate how the granularity of POS tags influences POS tagging, and furthermore, how POS tagging performance relates to parsing results. For this, we use the standard "pipeline" approach, in which a parser builds its output on previously tagged input. The experiments are performed on two German treebanks, using three POS tagsets of different granularity, and six different POS taggers, together with the Berkeley parser. Our findings show that less granularity of the POS tagset leads to better tagging results. However, both too coarse-grained and too fine-grained distinctions on POS level decrease parsing performance.

1 Introduction

German is a non-configurational language with a moderately free word order in combination with a case system. The case of a noun phrase complement generally is a direct indicator of the phrase's grammatical function. For this reason, a morphological analysis seems to be a prerequisite for a syntactic analysis. However, in computational linguistics, parsing was developed for English without the use of morphological information, and this same architecture is used for other languages, including German (Kübler et al., 2006; Petrov and Klein, 2008). An easy way of introducing morphological information into parsing, without modifying the architecture, is to attach morphology to the part-of-speech (POS) tagset. However, this makes POS tagging more complex and thus more difficult.

In this paper, we investigate the following questions: 1) How well do the different POS taggers work with tagsets of a varying level of morphological granularity? 2) Do the differences in POS tagger performance translate into similar differences in parsing quality? Complementary POS tagging results and preliminary parsing results have been published in German in Kübler and Maier (2013).

Our experiments are based on two different treebanks for German, TiGer (Brants et al., 2002) and TüBa-D/Z (Telljohann et al., 2012). Both treebanks are based on the same POS tagset, the Stuttgart-Tübingen Tagset (STTS) (Schiller et al., 1995). We perform experiments with three variants of the tagset: The standard STTS, the Universal Tagset (UTS) (Petrov et al., 2012) (a language-independent tagset), and an extended version of the STTS that also includes morphological information from the treebanks (STTSmorph). STTS consists of 54 tags, UTS uses 12 basic tags, and the morphological variants of the STTS comprise 783 and 524 POS tags respectively. We use a wide range of POS taggers, which are based on different strategies: Morfette (Chrupala et al., 2008) and RF-Tagger (Schmid and Laws, 2008) are designed for large morphological tagsets, the Stanford tagger (Toutanova et al., 2003) is based on a maximum entropy model, SVMTool (Giménez and Màrquez, 2004) is based on support vector machines, TnT (Brants, 2000) is a Markov model trigram tagger, and Wapiti (Lavergne et al., 2010) a conditional random field tagger. For our parsing experiments, we use the Berkeley parser (Petrov and Klein, 2007b; Petrov and Klein, 2007a).

Our findings for POS tagging show that Morfette reaches the highest accuracy on UTS and overall on unknown words while TnT reaches the best performance for STTS and the RF-Tagger for STTSmorph. These trends are stable across both treebanks. As for the parsing results, using STTS results in the best accuracies. For TiGer, POS tags assigned by the parser perform better in combination with UTS and STTSmorph. For TiGer in combination with STTS and all variants in TüBa-D/Z, there are only minor differences between the parser assigned POS tags and those by TnT.

The remainder of the article is structured as follows. In section 2, we review previous work. Section 3 presents the different POS tagsets. Section 4 describes our experimental setup. The POS tagging and parsing results are discussed in the sections 5 and 6, respectively. Section 7 concludes the article.

2 Previous Work

In this section, we present a review of the literature that has previously examined the correlation of POS tagging and parsing under different aspects. While this overview is not exhaustive, it presents the major findings related to our work. The issues examined can be regarded under two orthogonal aspects, namely, the parsing model used (data-driven or grammar-based), and the question of how to disambiguate between various tags for a single word.

Some work has been done on investigating different tagsets for individual languages. Collins et al. (1999) adapt the parser of Collins (1999) for the Czech Prague Dependency Treebank. Using an external lexicon to reduce data sparseness for word forms did not result in any improvement, but adding case to the POS tagset had a positive effect. Seddah et al. (2009) investigate the use of different parsers on French. They also investigate two tagsets with different granularity and come to the conclusion that the finer grained tagset leads to higher parser performance. The work that is closest to ours is work by Marton et al. (2013), who investigate the optimal POS tagset for parsing Arabic. They come to the conclusion that adding definiteness, person, number, gender, and lemma information to the POS tagset improve parsing accuracy. Both Dehdari et al. (2011) and Szántó and Farkas (2014) investigate automatic methods for selecting the best subset of morphological features, the former for Arabic, the latter for Basque, French, German, Hebrew, and Hungarian. However, note that Szántó and Farkas (2014) used the data from the SPMRL shared task 2013, which does not contain grammatical functions in the syntactic annotations. Both approaches found improvements for subsets of morphological features.

Other works examine, also within a "pipeline" method, possibilities for ambiguity reduction through modification of tagsets, or of the lexicon by tagset reduction, or through word-clustering. Lakeland (2005) uses lexicalized parsing à la Collins (1999). Similarly to the more recent work by Koo et al. (2008) or Candito and Seddah (2010), he addresses the question of how to optimally disambiguate for parsing on the lexical level by clustering. A word cluster is thereby seen as an equivalence class of words and assumes to a certain extent the function of a POS tag, but can be adapted to the training data. Le Roux et al. (2012) address the issue of data sparseness on the lexical level with PCFG parsing with the morphologically rich language Spanish. The authors use a reimplementation of the Berkeley parser. They show that parsing results can be improved by simplifying the POS tagset, as well as by lemmatization, since both approaches reduce data sparseness.

As already mentioned, a POS tag can be seen as an equivalence class of words. Since in the "pipeline" approach, the parse tree is built on POS tags, it is possible that a POS tagset is optimal from a linguistic point of view, but that its behavior is not optimal with respect to parsing results, because relevant lexical information is hidden from the parse tree by the POS tagset. While Koo et al. (2008) overcome this deficit by automatically searching for "better" clusters, other works copy certain lexical information into the actual tree, e.g., by using grammatical function annotation (Versley, 2005; Versley and Rehbein, 2009). Seeker and Kuhn (2013) complement the "pipeline" model (using a dependency parser (Bohnet, 2010)) by an additional component that uses case information as a filter for the parser. They achieve improvements for Hungarian, German and Czech.

A number of works develop models for simultaneous POS tagging or morphological segmentation and parsing. Based on work by Ratnaparkhi (1996) and Toutanova and Manning (2000), Chen and Kit (2011) investigate disambiguation on the lexical level. They assume that local, i.e., sequential but not

tag	description	tag	description	tag	description
NOUN	noun	PRON	pronoun	CONJ	conjunction
VERB	verb	DET	determiner, article	PRT	particle
ADJ	adjective	ADP	preposition, postposition	.	punctuation
ADV	adverb	NUM	numeral	X	everything else

Table 1: The 12 tags of the Universal Tagset.

hierarchical, features are decisive for the quality of POS tagging and note that a "pipeline" model does not take this into account since the parser effectively performs the POS disambiguation. On these grounds, they present a factorized model for PCFG parsing which separates parsing into a discriminative lexical model (with local features) and the actual parsing model, to be combined with a *product-of-experts* (Hinton, 1999).

Particularly in the dependency parsing literature, combined models for simultaneous POS tagging and parsing can be found. Research has concentrated on languages that require additional segmentation on the word level, such as Chinese (Hatori et al., 2011) or Hebrew (Goldberg and Tsarfaty, 2008). A new approach by Bohnet and Nivre (2012) was also evaluated on German. Results for POS tagging and parsing of German by means of a constraint grammar can be found in Daum et al. (2003) as well as in Foth et al. (2005). However, since these approaches are only marginally related to our approach, we forego a further overview.

3 The Three Tagset Variants

In our experiments, we use three POS tagset variants: The standard Stuttgart-Tübingen Tagset (STTS), the Universal Tagset (UTS) (Petrov et al., 2012), and an extended version of the STTS that also includes morphological information from the treebanks (STTSmorph). Since the two treebanks differ in their morphological annotation, in this variant, the tags differ between the two treebanks: For TiGer, we have 783 possible complex POS tags, and for TüBa-D/Z, there are 524. By complex tags, we mean a combination of an STTS tag with the morphological tag. Also, note that not all of the possible combinations are attested in the treebanks.

The UTS consists of 12 basic POS tags, shown in table 1[1]. It was developed for multilingual applications, in which a common tagset is of importance, such as for a multilingual POS tagger. The UTS only represents the major word classes. Thus, this tagset should result in a high POS tagging accuracy since only major distinctions are made. However, it is unclear whether these coarse distinctions provide enough information for a syntactic analysis.

The STTS is based on distributional regularities of German. It contains 54 tags and thus models more fine grained distinctions than the UTS. For a list of tags, see Schiller et al. (1995). The finer distinctions in STTS mostly concern word classes, but there is also a distinction between finite and infinite verbs. This distinction is important for the syntactic analysis, especially in TüBa-D/Z, but it can be difficult to make by a POS tagger with a limited context.

The STTS can be extended by a morphological component. Both treebanks provide a morphological analysis, but the analyses model different decisions. In TiGer, a set of 585 different feature combinations is used, which can be combined from the features listed in table 2. The sentence in (1) gives an example of the combination of the STTS and morphology, which are separated by the % sign. The feature – means that there are no morphological features for the given POS tag.

(1) Konzernchefs lehnen den Milliardär als
NN%Nom.Pl.Masc VVFIN%3.Pl.Pres.Ind ART%Acc.Sg.Masc NN%Acc.Sg.Masc APPR%–
US-Präsidenten ab /
NN%Acc.Sg.Masc PTKVZ%– $(%–
'Corporate CEOs disapprove of the billionaire as US president /'

[1]For a mapping from STTS to UTS, cf. `https://code.google.com/p/universal-pos-tags/`.

feature	description
ambiguous:	*
gender	masculine (Masc), feminine (Fem), neuter (Neut)
gradation	positive (Pos), comparative (Comp), superlative (Sup)
case	nominative (Nom), genitive (Gen), dative (Dat), accusative (Akk)
mode	indicative (Ind), conjunctive (Subj), imperative (Imp)
number	singular (Sg), plural (Pl)
person	1., 2., 3.
tense	present (Pres), past (Past)

Table 2: The morphological categories in TiGer.

feature	description
ambiguous	*
gender	masculine (m), feminine (f), neuter (n)
case	nominative (n), genitive (g), dative (d), accusative (a)
number	singular (s), plural (p)
person	1., 2., 3.
tense	present (s), past (t)
mode	indicative (i), conjunctive (k)

Table 3: The morphological categories in TüBa-D/Z.

Out of the 585 possible combinations of morphological features, 271 are attested in TiGer. In combination with the STTS, this results in 783 combinations of STTS and morphological tags. Out of those, 761 occur in the training set. However, we expect data sparseness during testing because of the high number of possible tags. For this reason, we calculated which percentage of the tags in the development and test set are known combinations. We found that 25% and 30%, respectively, do not occur in the training set. However, note that the number of tags in the development and test sets is considerably smaller than the number of tags in the training set.

In TüBa-D/Z, there are 132 possible morphological feature combinations which can be combined from the features listed in table 3. The sentence in (2) gives an example of the combination of the STTS and morphology.

(2) Aber Bremerhavens AfB fordert jetzt Untersuchungsausschuß
KON%– NE%gsn NE%nsf VVFIN%3sis ADV%– NN%asm
'But the Bremerhaven AfB now demands a board of inquiry'

Out of the 132 possible feature combinations, 105 are attested in TüBa-D/Z. In combination with the STTS, this results in 524 combinations of STTS and morphological tags. Out of those, 513 occur in the training set. For the development and test set, we found that 16% and 18% respectively do not occur in the training set. These percentages are considerably lower than the ones for TiGer.

Since the tagsets that include morphology comprise several hundred different POS tags, we expect tagging to be more difficult, resulting in lower accuracies. We also expect that the TüBa-D/Z tagset is better suited for POS tagging than the TiGer set because of its smaller tagset size and its higher coverage on the development and test set. It is, however, unknown whether this information can be used successfully in parsing.

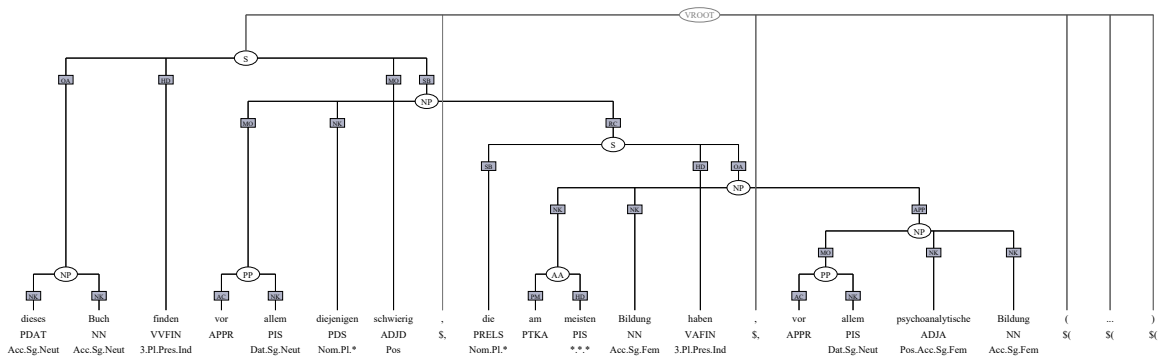

Figure 1: A sentence from TiGer.

4 Experimental Setup

4.1 Treebanks

We use the treebanks TiGer (Brants et al., 2002), version 2.2, and TüBa-D/Z (Telljohann et al., 2012), release 8. Both are built on newspaper text, *Frankfurter Rundschau* for TiGer and *taz* for TüBa-D/Z. Both treebanks use the same POS tagset with only one minor difference in the naming of one POS label. However, the treebanks differ considerably in the syntactic annotation scheme. While TiGer uses a very flat annotation involving crossing branches, the annotations in TüBa-D/Z are more hierarchical, and long distance relations are modeled via grammatical function labels rather than via attachment. Figures 1 and 2 show examples.

For preprocessing, we follow the standard practices from the parsing community. In both treebanks, punctuation and other material, such as parentheses, are not included in the annotation, but attached to a virtual root node. We attach the respective nodes to the tree using the algorithm described by Maier et al. (2012) so that every sentence corresponds to exactly one tree. In a nutshell, this algorithm uses the left and right terminal neighbors as attachment targets. In TiGer, we then remove the crossing branches using a two-stage process. In a first step, we apply the transformation described by Boyd (2007). This transformation introduces a new non-terminal for every continuous block of a discontinuous constituent. We keep a flag on each of the newly introduced nodes that indicates if it dominates the head daughter of the original discontinuous node. Subsequently, we delete all those nodes for which this flag is false.[2]

For both POS tagging and parsing, we use the same split for training, development, and test. We use the first half of the last 10 000 sentences in TiGer for development and the second half for testing. The remaining 40 472 sentences are used for training. Accordingly, in order to ensure equal conditions, we use the first 40 472 sentences in TüBa-D/Z for training, and the first and second half of the following 10 000 sentences for development and testing. The remaining sentences in TüBa-D/Z are not used.

4.2 POS Taggers

We employ six different POS tagger, each of them using a different tagging technique. Morfette (Chrupala et al., 2008), in its current implementation based on averaged Perceptron, is a tool designed for the annotation of large morphological tagsets. Since none of the other POS taggers have access to lemmas, we only provide full word forms to Morfette as well, which may inhibit its generalization capability. The RF-Tagger (Schmid and Laws, 2008) assumes a tagset in a factorized version. I.e., the POS tag VVFIN%3sis in sentence (2) would be represented as VVFIN.3.s.i.s, where the dots indicate different subcategories, which are then treated separately by the POS tagger. It is based on a Markov model, but the context size is determined by a decision tree. The Stanford tagger (Toutanova et al., 2003) is based on a maximum entropy model, and SVMTool (Giménez and Màrquez, 2004) is based on support vector machines. TnT (Brants, 2000; Brants, 1998), short for trigrams and tags, is a Markov model POS tagger.

[2] An implementation of all transformations is available at `http://github.com/wmaier/treetools`.

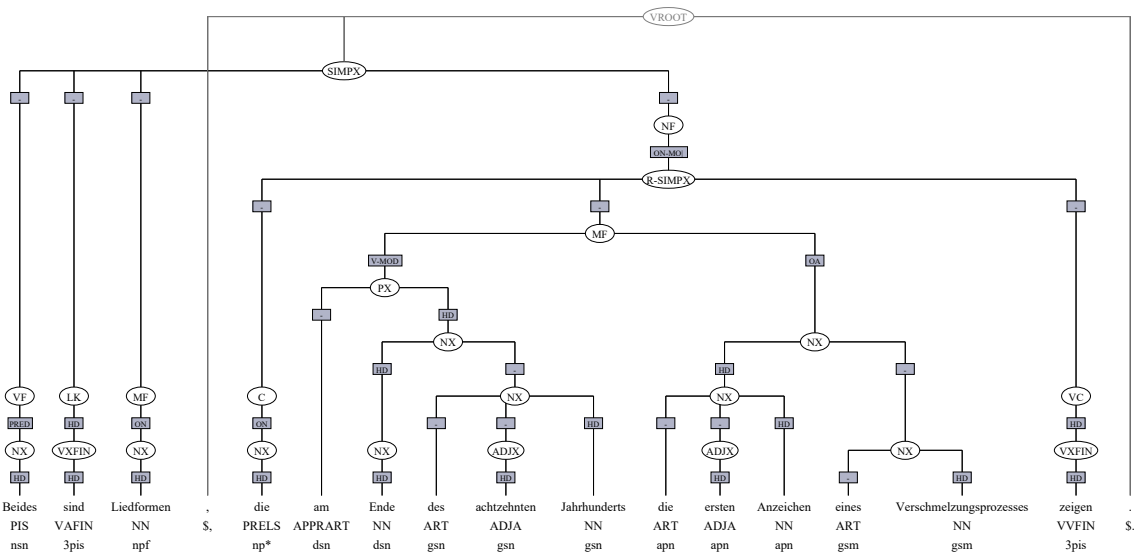

Figure 2: A sentence from TüBa-D/Z.

It uses an interpolation between uni-, bi- and trigrams as probability model. TnT has a sophisticated mechanism for tagging unknown words. We also use Wapiti (Lavergne et al., 2010) a conditional random field tagger. Since conditional random fields were developed for sequence tagging, this POS tagger is expected to perform well.

All POS taggers are used with default settings. For the Stanford tagger, we use the bi-directional model based on a context of 5 words. For SVMTool, we use the processing from left to right in combination with features based on word and POS trigrams and word length, prefix and suffix information. Wapiti is trained on uni-, bi-, and trigrams. Features used in training consist of tests concerning the alphanumeric, upper or lower case characteristics, prefixes and suffixes of length three, and all possible POS tags for a word.

For POS tagging evaluation, we use the script provided by TnT since it also allows us to calculate accuracy on known and unknown words.

4.3 Parser

We use the Berkeley parser (Petrov and Klein, 2007b; Petrov and Klein, 2007a). We chose the Berkeley parser because we are aware of the fact that there are considerable differences in the tagset sizes, which a plain PCFG parser cannot process successfully. The Berkeley parser split/merge capabilities provide a way of smoothing over these differences. For parser evaluation, we use our own implementation of the PARSEVAL metrics.[3] We report labeled precision (LP), labeled recall (LR), and the labeled F-score(LF1). Note that the labeled evaluation does not only look at constituent labels but also at grammatical functions attached to the constituents, e.g. NP-SBJ for a subject NP. This is a considerably more difficult task for German because of the relatively free word order. We also provide POS tagging accuracy in the parse trees since the Berkeley parser adapts POS tags from the input if they do not fit its syntax model.

5 POS Tagging Results

5.1 The Three Tagset Variants

The results for the POS tagging evaluation are shown in table 4. We are aware of the fact that the results are not directly comparable across the different POS tagsets and across different treebanks since the

[3]The implementation is available at `http://github.com/wmaier/evalb-lcfrs`. Note that we evaluate the trees as they are, i.e., we do not collapse or ignore tags.

Tagset	Tagger	TiGer dev	TiGer test	TüBa-D/Z dev	TüBa-D/Z test
UTS	Morfette	98.51	**98.09**	**98.25**	**98.49**
	RF-Tagger	97.89	97.41	97.69	97.96
	Stanford	97.88	96.83	97.11	97.26
	SVMTool	**98.54**	98.01	98.09	98.28
	TnT	97.94	97.48	97.72	97.92
	Wapiti	97.54	96.67	97.47	97.80
STTS	Morfette	94.12	93.23	92.95	93.41
	RF-Tagger	97.04	96.24	96.68	96.84
	RF-Tagger (fact.)	97.05	96.26	96.69	96.85
	Stanford	96.26	95.15	95.63	95.79
	SVMTool	97.06	96.22	96.46	96.69
	TnT	**97.15**	**96.29**	**96.92**	**97.00**
	Wapiti	92.93	91.62	90.99	91.81
STTSmorph	Morfette	82.71	80.10	81.19	82.26
	RF-Tagger	**86.56**	**83.90**	**85.68**	**86.31**
	Stanford	–	–	–	–
	SVMTool	82.47	79.53	80.33	81.31
	TnT	85.77	82.77	84.67	85.45
	Wapiti	79.83	75.92	77.27	78.29
STTSmorph → STTS	TnT	97.08	96.15	96.78	96.82

Table 4: POS tagging results using three versions of the German POS tagset and two treebanks.

corresponding tagging tasks differ in the level of difficulty. Any interpretation must therefore be taken with a grain of salt, but we think that it is important to evaluate POS tagging on its own, especially since it is not always the case that a larger label set automatically results in a more difficult task. The results show that UTS, i.e., the variant with the least information, results in the highest POS tagging results, between 96.67% and 98.54%. In tagging with the STTS, we reach a lower accuracy between 90.99% and 97.15%. When we include the morphological information, we reach considerably lower results, between 75.92% and 86.56%. In other words, this shows that the more information there is in the POS tagset, the harder the POS tagging task is. POS tagging with morphological information is the most difficult task. We also see that there are no results for the Stanford POS tagger in the morphological setting. We were unable to run these experiments, even when we used a high-memory cluster with access to 120g of memory. It seems that the Stanford tagger is incapable of handling the large tagset sizes in the setting using morphological information. Additionally, our assumption that the morphological tagset of TüBa-D/Z is less difficult to annotate because of its smaller tagset size is not borne out. The variation of results on TüBa-D/Z is often less than between the treebanks, across POS taggers.

If we compare the result of the different POS taggers, we see that for the different tagset variants, different POS taggers perform best: For UTS, surprisingly, Morfette reaches the highest results, with the exception of the TiGer development set, for which SVMTool performs slightly better. In general, SVMTool is very close in accuracy to Morfette for this tagset variant. For STTS, TnT outperforms all other POS taggers, and SVMTool is a close second. For STTSmorph, the RF-Tagger reaches the highest results. For the RF-Tagger in combination with the STTS, we performed 2 experiments, one using the standard STTS and one in which the STTS tags are factored, such that VVFIN is factored into V.V.FIN. The latter variant reaches minimally higher results. In all settings, Wapiti is the weakest approach; the difference between Wapiti and the best performing POS tagger reaches 6-7 percent points for STTSmorph. This is rather surprising given that POS tagging is a typical sequence tagging task, for which CRFs were developed.

Another fact worth mentioning is that there are considerable differences in POS tagging accuracy

		TiGer				TüBa-D/Z			
		dev		test		dev		test	
Tagset	Tagger	Known	Unkn.	Known	Unkn.	Known	Unkn.	Known	Unkn.
UTS	Morfette	98.66	**96.74**	98.32	**96.04**	98.54	**95.46**	98.69	**96.39**
	RF-Tagger	98.15	94.64	97.82	93.65	98.28	92.02	98.35	93.85
	Stanford	**99.05**	91.85	**98.78**	87.70	**98.94**	79.30	**98.92**	79.69
	SVMTool	98.81	95.26	98.41	94.45	98.63	92.89	98.66	94.27
	TnT	98.06	96.50	97.67	95.74	98.07	94.28	98.25	95.25
	Wapiti	98.94	80.71	98.51	80.04	98.68	85.79	98.83	86.91
STTS	Morfette	94.42	**90.60**	93.56	**90.24**	93.17	**90.83**	93.59	**91.57**
	RF-Tagger	97.80	87.92	97.30	86.71	97.62	87.59	97.73	87.52
	RF-T. (fact.)	97.78	88.21	97.28	87.09	97.63	87.65	97.73	87.51
	Stanford	**98.16**	73.56	**97.75**	71.60	**97.96**	73.04	**97.97**	72.64
	SVMTool	97.86	87.41	97.26	86.82	97.50	86.47	97.60	87.05
	TnT	97.80	89.25	97.21	87.95	97.65	89.78	97.72	89.33
	Wapiti	94.51	73.78	93.48	74.83	93.21	69.45	93.71	71.74
STTSmorph	Morfette	84.30	63.50	82.43	58.98	82.91	64.53	83.95	64.42
	RF-Tagger	**88.34**	**65.09**	**86.38**	**61.47**	**87.70**	**66.20**	**88.25**	**65.80**
	SVMTool	84.67	55.89	82.40	53.58	82.87	55.81	83.61	57.01
	TnT	87.62	63.41	85.55	57.65	86.91	62.95	87.61	62.55
	Wapiti	83.91	30.51	81.43	26.08	82.05	31.05	82.83	30.29

Table 5: Results for the different POS taggers for known and unknown words.

between the development and test set in both treebanks. For both STTS variants, these differences are often larger than the differences between individual POS taggers on the same data set. Thus, in the STTSmorph setting, the difference for TnT between the development and test set in TiGer is 3 percent points while the differences between TnT and SVMTool and Morfette respectively are less.

One last question that we investigated concerns the effect of the morphological information on POS tagging accuracy. We know that when we use morphological information, the POS tagging task is more difficult. However, it is possible that the mistakes that occur concern only the morphological information while the POS tags minus morphology may be predicted with equal or even higher accuracy. In order to investigate this problem, we used the STTSmorph output of TnT and deleted all the morphological information, thus leaving only the STTS POS tags. We then evaluated these POS tags against the gold STTS tags. The results are shown in the last row in table 4, marked as STTSmorph $\rightarrow$ STTS. A comparison of these results with the TnT results for STTS shows that the POS tagger reaches a higher accuracy when trained directly on STTS rather than on STTSmorph, with a subsequent deletion of the morphological information. This means that the morphological information is not useful but rather harmful in POS tagging.

5.2 Evaluating on Known and Unknown Words

In a next set of experiments, we investigate how the different POS taggers perform on known and unknown words. We define all words from the development and test set as known if the appear in the training set. If they do not, they are considered unknown words. Note, however, that even if a word is known, we still may not have the full set of POS tags in its ambiguity set. This is especially relevant for the larger tagsets where the ambiguity rate per word is higher.

In TiGer, 7.64% of the words in the development set are unknown, 9.96% in the test set. In TüBa-D/Z, 9.36% of the words in the development set are unknown, 8.64% in the test set. Note that this corresponds to the levels of accuracy in table 4.

The results of the evaluation on known and unknown words are shown in table 5. These results show that the Stanford POS tagger produces the highest accuracies for known words for UTS and STTS (note

	TiGer		TüBa-D/Z	
Morphology	dev	test	dev	test
STTS	**97.15**	**96.29**	**96.92**	**97.00**
STTSmorph	85.77	82.77	84.67	85.45
agreement	86.04	83.08	84.96	85.77
case	88.10	86.47	87.48	87.91
number	95.60	94.19	95.24	95.41
number + person	95.55	94.11	95.18	95.24
verbal features	97.03	96.02	96.55	96.44

Table 6: The results for TnT with different morphological variants.

that it could not be used for STTSmorph). For unknown words, Morfette reaches the highest results for UTS and STTS, with TnT reaching the second highest results. For STTSmorph, the RF-Tagger reaches the highest accuracy on both known and unknown words. The results for the RF-Tagger for STTS show that the factored version performs better on unknown words than the standard one. It is also noticeable that Wapiti, the CRF POS tagger, has the lowest performance on unknown words: For UTS, the results are 10-16 percent points lower that the ones by Morfette; for STTS, the difference reaches 16-23 percent points, and for STTSmorph, about 35 percent points. This shows that in order to reach a reasonable accuracy rate, Wapiti's unknown word handling model via regular expressions must be extended further. However, note that Wapiti's results on known words are also lower than the best performing system's, thus showing that CRFs are less well suited for POS tagging than originally expected.

5.3 Evaluating Morphological Variants

In this set of experiments, we investigate whether there are subsets of STTSmorph that are relevant for parsing and that would allow us to reach higher POS tagging and parsing accuracies than on the full set of morphological features. The subsets were chosen manually to model our intuition on which features may be relevant for parsing. We investigate the following subsets: all agreement features, case only, number only, number + person, and only verbal features. In this set of experiments, we concentrate on TnT because it has been shown to be the most robust across the different settings. The results of these experiments are shown in table 6. For comparison, we also list the results for the original STTS and STTSmorph settings from table 4.

The results show that there are morphological subsets that allow reliable POS accuracies: If we use verbal features, we reach results that are only slightly below the STTS results. For the subset using number + person features, the difference is around 2 percent points. However, all subsets perform worse than the STTS. The subsets that include case or all agreement features, which are the subsets most relevant for parsing, reach accuracies that are slightly above STTSmorph, but still more than 10 percent points below the original STTS.

6 Parsing Results

In this section, we report parsing results for TiGer in table 7 and for TüBa-D/Z in table 8. We again use the three POS tag variants as input, and we report results for 1) gold POS tags, 2) for tags assigned by TnT, which proved to be the most reliable POS tagger across different settings, and 3) for POS tags assigned by the Berkeley parser. Since the parser is known to alter POS tags given as input if they do not fit the syntax model, we also report POS tagging accuracy. Note that this behavior of the parser explains why we do not necessarily have a 100% POS tagging accuracy in the gold POS tag setting.

A first glance at the POS tagging results in the gold POS setting in tables 7 and 8 shows that for UTS and STTS, the decrease in accuracy is minimal. In other words, the parser only changes a few POS tags. When we compared the differences in POS tags between the output of the parser and the gold standard, we found that most changes constitute a retagging of common nouns (NN) as proper nouns (NE). In the STTSmorph setting, POS tagging accuracy is considerably lower, showing that the parser changed

		dev				test			
Tag source	Tagset	POS	LP	LR	LF1	POS	LP	LR	LF1
gold	UTS	100.00	77.97	77.23	77.60	99.97	71.80	70.26	71.02
	STTS	99.98	78.09	77.55	**77.82**	99.97	71.90	71.11	**71.50**
	STTSmorph	91.67	74.72	75.21	74.97	88.70	67.68	67.99	67.83
parser	UTS	**98.55**	77.75	76.84	77.29	**97.83**	71.13	69.50	70.30
	STTS	97.25	78.03	77.19	**77.60**	96.18	71.16	69.84	**70.49**
	STTSmorph	83.06	75.53	75.24	75.39	79.05	67.67	67.02	67.34
TnT	UTS	96.56	74.16	73.28	73.72	96.01	68.37	66.78	67.57
	STTS	97.26	78.03	77.19	**77.60**	96.19	71.16	69.84	**70.49**
	STTSmorph	77.94	73.06	72.69	72.88	75.05	65.43	64.78	65.10

Table 7: Parsing results for TiGer.

		dev				test			
Tag source	Tagset	POS	LP	LR	LF1	POS	LP	LR	LF1
gold	UTS	99.98	81.39	81.12	81.26	99.98	82.24	81.94	82.09
	STTS	100.00	83.60	83.58	**83.59**	99.99	84.54	84.46	**84.50**
	STTSmorph	89.75	82.27	78.85	80.53	90.55	83.57	79.91	81.70
parser	UTS	**98.35**	79.97	79.61	79.79	**98.58**	81.07	80.66	80.87
	STTS	97.20	81.84	81.65	**81.74**	97.39	82.93	82.78	**82.85**
	STTSmorph	81.03	80.85	77.22	78.99	81.68	81.89	78.20	80.00
TnT	UTS	**98.35**	79.97	79.61	79.79	**98.58**	81.07	80.66	80.87
	STTS	97.21	81.84	81.65	**81.74**	97.39	82.93	82.78	**82.85**
	STTSmorph	81.03	80.85	77.22	78.99	81.68	81.89	78.20	80.00

Table 8: Parsing results for TüBa-D/Z.

between 8% (UTS) and 25% (STTSmorph) of the POS tags. This is a clear indication that the parser suffers from data sparseness and has to adapt the POS tags in order to be able to parse the sentences.

We need to compare the POS tagging results based on automatically assigned POS tags; they show the following trends: For TiGer in the STTS setting, the results based on TnT and on the parser are very similar. For UTS and STTSmorph, the POS tags assigned by the parser reach a higher accuracy. For TüBa-D/Z, all the results are extremely similar.[4] If we compare the POS tagging accuracies of the parsed sentences and the accuracies of the original POS tags assigned by the tagger, we see that for TiGer, the accuracy decreases by approximately 1.5 percent points for UTS, 0.1 percent points for STTS and 9 percent points for STTSmorph. For TüBa-D/Z, the loss in the STTSmorph setting is smaller, at around 4 percent points. For UTS and STTS, there is a small improvement in POS tagging accuracy.

When we look at the parsing results, we see that gold POS tags always lead to the highest parsing results, across treebanks and POS tagsets. We also see that across all conditions, the parsing results for STTS are the highest. For TiGer, the results for UTS are only marginally lower, which seems to indicate that some of the distinctions made in STTS are important, but not all of them. For TüBa-D/Z, the loss for UTS is more pronounced, at around 2 percent points. This suggests that for the TüBa-D/Z annotation scheme, the more fined grained distinctions in STTS are more important than for UTS. One example would be the distinction between finite and infinite verbs, which is directly projected to the verb group in TüBa-D/Z (see the verb groups VXFIN and VXINF in figure 2). Note also that for Tüba-D/Z, the parsing based on automatic POS tagging outperforms parsing based on gold UTS tags, thus again confirming how important the granularity of STTS is for this treebank.

When we look at the parsing results for STTSmorph, it is obvious that this POS tagset variant leads to the lowest parsing results, even in the gold POS setting. This means that even though agreement

[4]Because of the (almost) identical results, we checked our results with extreme care but could not find any errors.

information should be helpful for assigning grammatical functions, the information seems to be presented to the parser in a form that it cannot exploit properly. We also performed preliminary experiments using the morphological variants discussed in section 5.3 in parsing, but the results did not improve over the STTS baseline.

When we compare the two sets of automatically assigned POS tags for TiGer, we see that the difference in POS accuracy for UTS is 1.8 percent points while the difference in F-scores is 2.5 percent points. This means that TnT tagging errors have a more negative impact on parsing quality than those in the POS tags assigned by the parser itself. For STTSmorph, the difference is more pronounced in POS accuracy (4 points as opposed to 2.2 in F-scores), which means that for STTSmorph, TnT errors are less harmful than for UTS. We assume that this is the case because in many instances, the POS tags themselves will be correct, and the error occurs in the morphological features. For TüBa-D/Z, the difference between UTS and STTSmorph is marginal; this is due to the fact that UTS results are much lower than for TiGer. Thus, the difference between STTS and STTSmorph is stable across both treebanks.

A more in-depth investigation of the results shows that the aggregate EVALB score tends to hide individual large differences between single sentences in the results. For example, in the results for the TiGer dev set with gold POS tags, there are 119 sentences in STTSmorph which have an STTS counterpart with an F-score that is at least 50 points higher. However, there are also 28 sentences for which the opposite holds, i.e., for which STTSmorph wins over STTS. In TüBa-D/Z, there are fewer sentences with such extreme differences. There are 28 / 11 sentences with a score difference of 50 points or more between STTS and STTSmorph in the TüBa-D/Z development set, and vice versa. A manual inspection of the results indicates that in some cases, the morphology is passed up into the tree and thereby contributes to a correct grammatical function of a phrase label (such as for case information) while in other cases, it causes an over-differentiation of grammatical functions and thereby has a detrimental effect (such as for PPs, which are attached incorrectly). In the case of TüBa-D/Z, this leads to trees with substructures that are too flat, while in the case of TiGer, it leads to more hierarchical substructures. This finding is corroborated by a further comparison of the number of edges produced by the parser, which reveals that for the case of TiGer, the number of edges grows with the size of the POS tagset, while for the case of TüBa-D/Z, the number of edges produced with STTS is higher than with UTS, but drops considerably for STTSmorph. The large differences in results for single sentences look more pronounced in TiGer due to the average number of edges per sentence (7.60/8.72 for dev/test gold), which is much lower than for TüBa-D/Z (20.93/21.16 for dev/test gold); in other words, because of its flat annotation. We suspect that there is data sparsity involved, but this needs to be investigated further.

7 Conclusion and Future Work

We have investigated how the granularity of POS tags influences POS tagging, and furthermore, how POS tagging performance relates to parsing results, on the basis of experiments on two German treebanks, using three POS tagsets of different granularity (UTS, STTS, and STTSmorph), and six different POS taggers, together with the Berkeley parser.

We have shown that the tagging task is easier the less granular the tagset is. Furthermore, we have shown that both too coarse-grained and too fine-grained distinctions on POS level hurt parsing performance. The results for the morphological tagset are thus in direct contrast to previous studies, such as (Dehdari et al., 2011; Marton et al., 2013; Seddah et al., 2009; Szántó and Farkas, 2014), which show for different languages that adding morphological information increases parsing accuracy. Surprisingly, given the STTS tagset, the Berkeley parser itself was able to deliver a POS tagging performance which was almost identical to the performance of the best tagger, TnT. Additionally, we can conclude that the choice of the tagset and of the best POS tagger for a given treebank does not only depend on the language but also on the annotation scheme.

In future work, we will undertake a systematic investigation of tag clustering methods in order to find a truly optimally granular POS tagset. We will also investigate the exact relation between annotation depth and the granularity of the POS tagset with regard to parsing accuracy and data sparsity. The latter may elucidate reasons behind the differences between our results and those of the studies mentioned above.

References

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1455–1465, Jeju Island, Korea.

Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (IJCNLP)*, pages 89–97, Beijing, China.

Adriane Boyd. 2007. Discontinuity revisited: An improved conversion to context-free representations. In *Proceedings of The Linguistic Annotation Workshop (LAW) at ACL 2007*, pages 41–44, Prague, Czech Republic.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT)*, pages 24–41, Sozopol, Bulgaria.

Thorsten Brants, 1998. *TnT–A Statistical Part-of-Speech Tagger*. Universität des Saarlandes, Computational Linguistics, Saarbrücken, Germany.

Thorsten Brants. 2000. TnT–a statistical part-of-speech tagger. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics and the 6th Conference on Applied Natural Language Processing (ANLP/NAACL)*, pages 224–231, Seattle, WA.

Marie Candito and Djamé Seddah. 2010. Parsing word clusters. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 76–84, Los Angeles, CA.

Xiao Chen and Chunyu Kit. 2011. Improving part-of-speech tagging for context-free parsing. In *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1260–1268, Chiang Mai, Thailand.

Grzegorz Chrupala, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with Morfette. In *Proceedings the Fifth International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco.

Michael Collins, Jan Hajič, Lance Ramshaw, and Christoph Tillmann. 1999. A statistical parser for Czech. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 505–512, College Park, MD.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Michael Daum, Kilian Foth, and Wolfgang Menzel. 2003. Constraint based integration of deep and shallow parsing techniques. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Budapest, Hungary.

Jon Dehdari, Lamia Tounsi, and Josef van Genabith. 2011. Morphological features for parsing morphologically-rich languages: A case of Arabic. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 12–21, Dublin, Ireland.

Kilian Foth, Michael Daum, and Wolfgang Menzel. 2005. Parsing unrestricted German text with defeasible constraints. In H. Christiansen, P. R. Skadhauge, and J. Villadsen, editors, *Constraint Solving and Language Processing*, pages 140–157. Springer.

Jesús Giménez and Lluís Màrquez. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 43–46, Lisbon, Portugal.

Yoav Goldberg and Reut Tsarfaty. 2008. A single generative model for joint morphological segmentation and syntactic parsing. In *Proceedings of The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*, pages 371–379, Columbus, OH.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2011. Incremental joint POS tagging and dependency parsing in Chinese. In *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1216–1224, Chiang Mai, Thailand.

Geoffrey Hinton. 1999. Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, pages 1–6, Stockholm, Sweden.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*, pages 595–603, Columbus, OH.

Sandra Kübler and Wolfgang Maier. 2013. Über den Einfluss von Part-of-Speech-Tags auf Parsing-Ergebnisse. *Journal for Language Technology and Computational Linguistics. Special Issue on "Das Stuttgart-Tübingen Wortarten-Tagset – Stand und Perspektiven"*, 28(1):17–44.

Sandra Kübler, Erhard W. Hinrichs, and Wolfgang Maier. 2006. Is it really that difficult to parse German? In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 111–119, Sydney, Australia.

Corrin Lakeland. 2005. *Lexical Approaches to Backoff in Statistical Parsing*. Ph.D. thesis, University of Otago, New Zealand.

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, Uppsala, Sweden.

Joseph Le Roux, Benoit Sagot, and Djamé Seddah. 2012. Statistical parsing of Spanish and data driven lemmatization. In *Proceedings of the ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*, pages 55–61, Jeju, Republic of Korea.

Wolfgang Maier, Miriam Kaeshammer, and Laura Kallmeyer. 2012. Data-driven PLCFRS parsing revisited: Restricting the fan-out to two. In *Proceedings of the Eleventh International Conference on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, Paris, France.

Yuval Marton, Nizar Habash, and Owen Rambow. 2013. Dependency parsing of Modern Standard Arabic with lexical and inflectional features. *Computational Linguistics*, 39(1):161–194.

Slav Petrov and Dan Klein. 2007a. Improved inference for unlexicalized parsing. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 404–411, Rochester, NY.

Slav Petrov and Dan Klein. 2007b. Learning and inference for hierarchically split PCFGs. In *Proceedings of AAAI (Nectar Track)*, Vancouver, Canada.

Slav Petrov and Dan Klein. 2008. Parsing German with language agnostic latent variable grammars. In *Proceedings of the ACL Workshop on Parsing German*, pages 33–39, Columbus, OH.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP)*, pages 133–142, Philadelphia, PA.

Anne Schiller, Simone Teufel, and Christine Thielen. 1995. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, Universität Stuttgart and Universität Tübingen.

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)*, pages 777–784, Manchester, UK.

Djamé Seddah, Marie Candito, and Benoît Crabbé. 2009. Cross parser evaluation: A French Treebanks study. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*, pages 150–161, Paris, France.

Wolfgang Seeker and Jonas Kuhn. 2013. Morphological and syntactic case in statistical dependency parsing. *Computational Linguistics*, 39(1):23–55.

Zsolt Szántó and Richárd Farkas. 2014. Special techniques for constituent parsing of morphologically rich languages. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 135–144, Gothenburg, Sweden.

Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck, 2012. *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Germany.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, Hong Kong.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 252–259, Edmonton, Canada.

Yannick Versley and Ines Rehbein. 2009. Scalable discriminative parsing for German. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*, pages 134–137, Paris, France.

Yannick Versley. 2005. Parser evaluation across text types. In *Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, Barcelona, Spain.