Is Part-of-Speech Tagging a Solved Task?
An Evaluation of POS Taggers for the German Web as Corpus

Eugenie Giesbrecht
FZI Research Center
for Information Technology
76131 Karlsruhe, Germany
giesbrecht@fzi.de

Stefan Evert
Institute of Cognitive Science
University of Osnabrck
49069 Osnabrück, Germany
stefan.evert@uos.de

Abstract

Part-of-speech (POS) tagging is an important preprocessing step in natural language processing. It is often considered to be a "solved task", with published tagging accuracies around 97%. Our evaluation of five state-of-the-art POS taggers on German Web texts shows that such high accuracies can only be achieved under artificial cross-validation conditions. In a real-life scenario, accuracy drops below 93% with enormous variation between different text genres, making the taggers unsuitable for fully automatic processing. We find that HMM taggers are more robust and much faster than advanced machine-learning approaches such as MaxEnt. Promising directions for future research are unsupervised learning of a tagger lexicon from large unannotated corpora, as well as developing adaptive tagging models.

1 Introduction

Automatic part-of-speech (POS) tagging is an important and widely-used preprocessing step in natural language processing applications, and it is almost indispensable for the exploitation of corpus data. At the same time, it is essentially considered a "solved task", with state-of-the-art taggers achieving per-word accuracies of 97%–98% (Schmid, 1995; Toutanova et al., 2003; Shen et al., 2007). While this still means that, on average, every other sentence contains a tagging error,[1] the accuracy is close to the level of agreement between human annotators and thus to the upper limit that can be expected from an automatic tagger.

Virtually all taggers have been trained and evaluated on newspaper text, though, and it is not clear whether they would achieve equally high accuracy on other genres such as spoken language, informal writing, or Web pages. The latter form a particularly important category in scientific research – where an increasing number of researchers turn to the World Wide Web as a convenient and inexhaustible source of natural language data (the "Web as Corpus" approach, see e.g. Kilgarriff and Grefenstette (2003)) – as well as commercial applications – where mining the Web for semantic knowledge, market intelligence, etc. has become one of the most successful applications of NLP technologies.

Therefore, the reported tagging accuracies of 97%–98% have to be understood as optimistic estimates, representing an ideal case for machine-learning approaches: (i) the taggers are applied to edited, highly standardized text with a low rate of errors and unusual patterns; and (ii) training and test data are very similar (usually from the same volume of the same newspaper), so that overfitting of the training data is rewarded to a certain degree.

The goal of this paper is to find out whether the published tagging accuracies – which are often taken for granted by researchers and developers using off-the-shelf POS taggers in their NLP systems – can also be achieved under real-life conditions, where taggers have to deal with less standardized genres such as Web texts. Our hypothesis is that the quality of POS tagging will be dramatically reduced under such circumstances, perhaps even to a degree that makes its usefulness as a general preprocessing step questionable.[2] In order to test this hypothesis, we evaluate five state-of-

[1] With a per-word tagging accuracy of 97%, there is a probability of 45.6% that a 20-word sentence (the average sentence length in the Brown corpus) contains one or more tagging errors.

[2] With a per-word accuracy of 92%, less than one in five sentences will be error-free. Some sources also claim that the baseline accuracy achieved by a simple most-frequent-tag heuristic can be as high as 90% under favourable conditions, cf. (Manning and Schütze, 1999, 372).

the-art statistical taggers on a representative collection of German Web texts sampled from the DEWAC corpus (Baroni et al., 2009). Since we are not aware of any systematic comparative evaluation of German POS taggers, we also determine "ideal" tagging accuracies by cross-validation on the TIGER treebank (Brants et al., 2002), to be used as a point of reference.

The rest of the paper is organized as follows. Section 2 gives an overview of the state of the art in statistical POS tagging and lists published evaluation results for German. Section 3 describes our evaluation methodology and the corpora used in our experiments. Evaluation results are given in Section 4, with a qualitative analysis of tagging errors in Section 5. Section 6 examines how tagging accuracy is influenced by tagset granularity and the genre of a Web page. The main insights we have obtained for the development of more robust POS taggers are summarized in Section 7.

2 State-of-the-art taggers for German

Most POS taggers have been developed for English, using the Penn Treebank (Marcus et al., 1993) as training and evaluation data. The best published tagging accuracies fall into a narrow range from 96.50% to 97.33% (Brants, 2000; Toutanova et al., 2003; Giménez and Màrquez, 2004; Shen et al., 2007). While the rule-based EngCG tagger is reported to achieve very high accuracy in combination with a statistical disambiguator (Tapanainen and Voutilainen, 1994), it is only available as a commercial product and has therefore been excluded from our study.

However, these high accuracy figures have to be qualified for two reasons. First, there are some doubts about the consistency of the Penn Treebank annotation (Dickinson and Meurers, 2003). Second, the proportion of unknown words is very low in all reported evaluation experiments (ca. 2%). It is not clear whether comparable results would be achieved for a text genre with richer, less controlled vocabulary (such as Web pages) or a language with more complex and productive morphology (such as German).

There are only few published evaluation results for German POS taggers, summarized in Table 1. The top two rows show accuracies reported by the developers of the two most widely-used statistical taggers for German, TnT (Brants, 2000) and TreeTagger (Schmid, 1995). Both are in the same range as state-of-the-art English taggers, and TreeTagger even outperforms the best current tagger for English. These results are not directly comparable, though, since they have been obtained on different gold standards – TnT was trained and evaluated on the NEGRA treebank (Skut et al., 1998), TreeTagger on a proprietary gold standard.

	overall	UW	KW	% unk.
TnT	96.70	89.0	97.7	11.9
TreeTagger	97.53	78.0	97.4	2.0
TBL	94.57	81.5	—	15.0
TreeTagger	95.27	84.1	—	15.0

Table 1: Published evaluation results of German POS taggers (UW = accuracy on unknown words, KW = accuracy on known words, % unk. = proportion of unknown words; all values are percentages). The top rows show results reported by the original developers (Brants, 2000; Schmid, 1995), the bottom rows show results from a comparative evaluation study (Volk and Schneider, 1998).

As expected, the proportion of unknown words (12%–15%) is much higher than for the English taggers. Note that TreeTagger makes use of a heuristic lexicon extracted from a large, automatically tagged corpus (Schmid, 1995, Sec. 3.3). This lexicon reduces the proportion of unknown words to only 2%, similar to the Penn Treebank, and is also included in the standard parameter file distributed with TreeTagger (cf. Sec. 3). When Volk and Schneider (1998) re-train the tagger without such a heuristic lexicon, the proportion of unknown words increases to 15%.

The bottom rows of Table 1 show results from an independent evaluation study (Volk and Schneider, 1998), comparing TreeTagger with Brill's (1995) transformation-based learning approach (TBL). The accuracy of TreeTagger is much lower than reported by Schmid (1995) – only 95.27% *vs* 97.53% – and falls behind the English state of the art. While differences in the training regime may account for part of the decrease, the most important factor is certainly the higher proportion of unknown words (15% *vs* 2%) resulting from the lack of a heuristic lexicon. Still, the statistical approach of TreeTagger outperforms the rule-based TBL tagger and is also computationally more efficient with a training time of less than 2 minutes *vs* approx. 30 hours for TBL (Volk and Schneider, 1998, Sec. 2).

2.1 Taggers selected for the evaluation

We decided to restrict our evaluation to statistical taggers, which achieve the best published results for both English and German. Likewise, only freely available implementations – which could easily be trained and evaluated on our data, and are most widely used by researchers and developers – were taken into consideration. In addition to the best-performing German taggers (TnT and TreeTagger), we included three further state-of-the-art taggers, resulting in the following list of candidates:

1. TreeTagger[3] – HMM tagger using decision trees for smoothing; best published tagging accuracy for German; widely used by researchers due to its easy availability (Schmid, 1995);
2. TnT – another widely-used HMM tagger, with standard smoothing (Brants, 2000);
3. SVMTagger – open-source tagger using support vector machines for classification (Giménez and Màrquez, 2004);
4. Stanford tagger – bidirectional MaxEnt tagger with the best published tagging accuracy for English (Toutanova et al., 2003);
5. Apache UIMA Tagger[4] – open-source HMM tagger written in Java, implemented by one of the authors (see below for details).

2.2 The UIMA Tagger

The UIMA Tagger closely follows the standard HMM approach described by Brants (2000), omitting some advanced heuristics that are used by the TnT implementation but not mentioned in the paper. Like TnT, the UIMA Tagger is based on a trigram Hidden Markov Model, with trigram probabilities smoothed by deleted interpolation. Lexical probabilities of unknown words are guessed from suffix strings, estimated from words that occur less than 10 times in the training corpus. Separate suffix probabilities are computed for captialized and non-capitalized words, since capitalization provides an important morphological cue in German (all common nouns are captialized).

For known words, only the tags available in the model are used for prediction; otherwise Ukkonen suffix trees (Ukkonen, 1995) are used to find the longest suffix of an unknown word for which a suffix probability has been estimated. No further heuristics and smoothing strategies are implemented in the current version of the UIMA Tagger.

The UIMA Tagger was included in our evaluation because it provides an excellent open-source platform for experiments on improving tagging accuracy, while other HMM taggers such as TnT and TreeTagger are only available in the form of binary packages. The recent open-source implementation HunPos (Halácsy et al., 2007) is written in OCaml, which has a much smaller user base than Java. Last but not least, the UIMA Tagger is natively supported in Apache UIMA[5] (Unstructured Information Management Architecture), a framework for industrial text analytics applications that is also being used by an increasing number of NLP and Information Retrieval researchers (Müller et al., 2008; Nyberg et al., 2008). Together with its permissive Apache License, this will encourage academic and industrial research groups to adapt the tagger to their special requirements (such as processing Web pages), and to contribute their improvements back to the open-source code base.

3 Evaluation methodology

Since no directly comparable evaluation results have been published for German POS taggers, we first evaluated all five taggers listed in Section 2.1 on the TIGER treebank (Brants et al., 2002), which is currently the largest manually annotated German corpus. It consists of about 900,000 tokens (50,000 sentences) of German newspaper text, taken from the *Frankfurter Rundschau*.[6] Each sentence has been annotated with manually validated POS tags, lemmas, morphosyntactic features and parse trees. Annotations were carried out by two independent annotators, followed by a consistency check (Brants and Hansen, 2002). For our purposes, only the POS annotation according to the STTS tagset (Schiller et al., 1999) was used.

The evaluation was carried out by 10-fold cross-validation. We divided the corpus into 10 contiguous parts, which we consider to be a slightly more realistic setting than taking every tenth sentence

[3] Binary packages for Linux, Solaris, Mac OS X and Windows can be downloaded from `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html`, together with pre-compiled parameter files for 8 different languages.

[4] The UIMA Tagger can be downloaded from `http://incubator.apache.org/uima/sandbox.html#tagger.annotator`

[5] `http://incubator.apache.org/uima/`

[6] The token counts given in this paper include all tokens, i.e. words, numbers and punctuation.

or choosing random sentences. Then, each tagger was trained on 9 of the 10 parts (using standard settings for all meta-parameters) and evaluated on the held-out part. In Section 4.1, we report the mean and standard deviation of per-word tagging accuracy across all 10 cross-validation folds.[7] This evaluation setup is very similar to published evaluation experiments for TnT, TreeTagger, and the English POS taggers. It provides a fair comparison of the five different taggers and serves as a point of reference for our main evaluation experiment on Web texts. One has to keep in mind, though, that – like in most other published evaluation studies – the POS taggers are evaluated on text that is very similar to their training data, which will rarely be the case in real-world applications.

Finally, all taggers are trained on the complete TIGER treebank. The resulting parameter files are used for all further evaluation experiments, ensuring a fair comparison between the taggers. In addition, we evaluate the standard parameter files (SPF) distributed with TnT and TreeTagger, which many researchers use for convenience.

Since no manually annotated Web reference corpus is available, we had to compile our own gold standard for the evaluation on Web text. For this purpose, we selected a random sample of Web pages from DEWAC (Baroni et al., 2009), a German Web corpus containing approx. 1.6 billion tokens of text collected in the year 2005.[8] The DEWAC corpus was cleaned by removing duplicate pages and so-called boilerplate (automatically generated page content such as navigation bars, advertising and legal disclaimers). It was then tagged and lemmatized using TreeTagger with its standard parameter file for German. See Baroni et al. (2009) for details on the corpus preparation.

Our gold standard consists of 13 Web pages from widely different genres, amounting to a total of 10,057 tokens of text. We manually corrected the automatic tokenization and POS tagging provided by the DEWAC corpus, using the same STTS tagset as in the TIGER treebank. In Section 4.2, we report the per-word accuracy achieved by each of the five taggers on the DEWAC gold standard, using the TIGER treebank for training (as well as SPF for TnT and TreeTagger). Since these results are not obtained through a cross-validation scheme, it is not meaningful to calculate standard deviation (but see Section 6.1 for the variability of tagging accuracy across text genres).

4 Evaluation results

4.1 TIGER treebank

The top row of Table 3 shows the mean and standard deviation of per-word tagging accuracy on the TIGER treebank for all selected taggers, obtained by 10-fold cross-validation as described in Section 3. The other rows give separate accuracy figures for known and unknown words, as well as the percentage of unknown words in the test data. Accuracies obtained with the standard parameter files of TnT and TreeTagger are shown in Table 2.[9]

Tagging with the standard parameter file of TreeTagger results in a per-word accuracy of 95.82%, which is 1.71% less than the value reported by Schmid (1995). The accuracy of TnT is also considerably lower than the published figure. In the cross-validation experiment (Table 3), where training and test data are from the same corpus, both taggers achieve considerably better accuracy, though TreeTagger still falls short of the published value of 97.53% (probably due to the lack of a heuristic lexicon in our experiments).

	overall	KW	UW	% unk.
TreeTagger	95.82	96.27	79.88	2.7
TnT	95.71	96.97	86.94	12.6

Table 2: Per-word tagging accuracy on the TIGER treebank, using the standard parameter files (SPF) distributed with TreeTagger and TnT.

The best result in the cross-validation experiment was achieved by the bidirectional MaxEnt Stanford tagger, whose mean total accuracy of 97.63% matches the published figure for TreeTagger, making it the best known POS tagger for German text. It is also remarkable that this total accuracy is as high as the known-words accuracy of TreeTagger and TnT. Second place is achieved by the SVM tagger. The Stanford tagger is significantly better than all HMM taggers (paired t-test against TnT/TreeTagger: t=11.33, df=9, p<.001) and the SVM tagger (paired t-test: t=13.21, df=9,

[7] Since all folds contain approximately the same number of tokens, this macro-averaged mean is equal to the micro-averaged per-word accuracy on the full corpus.

[8] Note that Baroni et al. (2009) report a much smaller size of approx. 1.28 billion tokens, because their counts exclude punctuation, numbers and other non-word tokens (Baroni et al., 2009, Sec. 3.4).

[9] Since these results have not been obtained by cross-validation, standard deviation is not available.

	TreeTagger	Stanford	UIMA	TnT	SVM
total accuracy (%)	96.89±0.34	**97.63±0.24**	96.04±0.38	96.92±0.31	97.12±0.20
known words (%)	97.62±0.21	–	97.50±0.18	97.59±0.20	97.71±0.17
unknown words (%)	87.89±0.99	**91.66±0.83**	79.59±1.30	89.16±0.85	90.16±0.84
% of unknown words	7.44±0.78	7.52±0.46	8.10±0.71	7.85±0.88	7.82±0.82

Table 3: Evaluation results for 10-fold cross-validation on the TIGER treebank. For each tagger, we report mean and standard deviation of per-word accuracy across the 10 folds (all values are percentages).

p<.001). This is mostly due to its significantly higher accuracy on unknown words.

The high accuracy of the Stanford tagger comes at a price, though, due to the computational complexity of its advanced statistical model. Tagging the 900,000 tokens of the TIGER treebank takes more than 45 minutes with the Stanford tagger, compared to less than 10 seconds with TreeTagger (measured on 2.6 GHz Dual Core AMD Opteron 285 Processor). Likewise, training the Stanford tagger on TIGER took approx. 5.5 hours, while the TreeTagger completed its supervised training procedure in less than 10 seconds. The gain in accuracy of approx. 0.7% compared to the best HMM tagger is relatively small, and it is presumably worth its while in case achieving the best possible accuracy is crucial for the task at hand.

4.2 Web texts

For this experiment, we trained all taggers on the complete TIGER treebank and then evaluated their performance on DEWAC, in order to simulate a realistic setting where no in-domain training data are available and a standard parameter file trained on a newspaper corpus has to be used. Evaluation results are shown in Table 4; the first column lists the results obtained by TreeTagger with its standard parameter file (labelled TT-SPF).

Disregarding the TT-SPF data, we see that the best overall accuracy is now achieved by TnT, a HMM-based tagger. While the Stanford tagger is considerably better than its competitors on unknown words, its overall accuracy falls slightly short of TnT.[10] These results clearly indicate a certain degree of overtraining for the machine-learning approaches (Stanford and SVM tagger), while TnT generalizes better to less standardized genres such as Web texts. We may thus conclude that HMM-based approaches are both more robust and computationally more efficient than MaxEnt and other advanced machine-learning techniques.

Surprisingly, TreeTagger performs worse than all other taggers if it is trained on the TIGER treebank; the reasons for this discrepancy are not entirely clear yet. When used with its standard parameter file (SPF), on the other hand, it achieves a much higher accuracy than TnT (93.71% *vs* 92.69%). This appears to be due to the inclusion of a heuristic tagger lexicon in the SPF, which reduces the proportion of unknown words to 4.15%, compared to 13.44% for TnT.

On the whole, there is a dramatic decrease in accuracy for all taggers under real-life conditions, caused (amongst others) by a much higher proportion of unknown words than in the cross-validation experiment. The unknown words in Web texts also seem to be more "difficult" than those in TIGER, so that e.g. the unknown-words accuracy of the Stanford tagger drops from 91.66% to 75.35%. The most robust results are achieved by TreeTagger with its standard parameter file, but a per-word accuracy of 93.71% is still unsatisfactory for most applications in linguistics and NLP.

5 Qualitative error analysis

A closer look at the error statistics for individual tags – using the best-performing tagger on DEWAC, i.e. TreeTagger with its SPF, as an example – revealed similar error sources as reported by Schmid (1995) and Volk and Schneider (1998). Most of the errors can be traced to insufficient distributional differences within major categories (e.g., proper *vs* common nouns or finite *vs* infini-

[10] It is difficult to determine whether the observed differences are significant, since these data have not been obtained from a cross-validation procedure. In view of the enormous variation between individual texts in the DEWAC gold standard (see discussion in Section 6.1), it is clearly inappropriate to pool all data into a sample of 10,057 tokens. Paired t-tests across the 13 individual texts find significant differences (wrt. macro-averaged accuracy as shown in Table 7) only between TnT and TreeTagger (as well as TT-SPF and TnT), again due to the large variation between texts.

	TT-SPF[a)]	TT[b)]	Stanford	UIMA	TnT	SVM
total accuracy (%)	**93.71**	90.78	92.61	91.68	**92.69**	92.36
known words (%)	95.42	93.59	—	95.59	95.90	95.91
unknown words (%)	54.30	69.12	**75.35**	66.49	71.99	69.45
% of unknown words	4.15	11.48	13.00	13.43	13.44	13.43

[a] TreeTagger with standard parameter file included in distribution
[b] TreeTagger with parameter file trained on the TIGER treebank

Table 4: Evaluation results on the DEWAC gold standard. All taggers have been trained on the complete TIGER treebank for this experiment (except for TT-SPF).

tive verbs) or between certain categories (e.g., adverbs *vs* adverbially used adjectives).

TIGER	DEWAC	TIGER	DEWAC
NE	**$(**	ADJD	AVD
APPR	NE	ADJA	**XY**
VVFIN	**FM**	PIS	**CARD**
ADV	NN	VVINF	ADJA
NN	VVFIN	VVPP	APPR

Table 5: Most frequently misclassified POS tags in TIGER and DEWAC (TreeTagger with SPF).

TIGER Treebank		DEWAC	
correct tag	TT-SPF	correct tag	TT-SPF
NE	NN	NE	NN
APPR	KOKOM	**$(**	**$.**
NN	NE	**FM**	**NN**
VVINF	VVFIN	NN	NE
VVFIN	VVPP	**FM**	**NE**
ADJA	NN	**CARD**	**NN**
PWAV	KOUS	**$(**	**ADJA**
ADV	ADJD	ADV	ADJD
ADJD	ADV	**XY**	**NE**
VVFIN	VVINF	VVFIN	VVPP

Table 6: Most frequently confused POS tags.

Table 5 shows the gold standard POS tags that were misclassified most frequently. Apart from typical tagging errors for the main parts of speech such as nouns and verbs, there are a number of unexpected tags among the 10 most frequent error types on Web texts: `$(` (sentence-internal punctuation, except for comma), `FM` (foreign material), `CARD` (cardinal numbers) and `XY` (special characters). All of these are prevalent in Web texts, and they appear to be an important factor behind the low tagging accuracy.

The comparison of the most frequent tag confusion pairs for TIGER and DEWAC (see Table 6) confirms our intuition that – in addition to well-known problems (Schmid, 1995; Volk and Schneider, 1998) that were confirmed by our TIGER experiments – there are many "new" error types due to the confusion of punctuation signs, foreign words and cardinals with common nouns, proper nouns and adjectives.

6 Determinants of tagging accuracy

6.1 Text genre

The Web pages included in our DEWAC gold standard represent entirely different text genres. This allowed us to test whether the low overall tagging accuracy in Table 4 reflects a general difficulty of processing Web data, or whether there are "easier" and "harder" genres on the Web. Table 7 shows separate per-word accuracy results for each genre.

In 7 out of 13 genres, TreeTagger with its standard parameter file (TT-SPF) achieves state-of-the-art accuracy between 95.42% and 98.25%. These "easy" genres include various news reports, a political speech, a support programme announcement, and other types of expository prose – all quite similar to typical newspaper text. In most cases, the percentage of unknown words is also very low (details omitted for space reasons).

Clearly, there are four problematic genres, where the accuracy of all taggers falls below 94%: an episode guide for a TV series, postings from an online forum, a conference information site,[11] and a news report on the archbishop of Boston (highlighted in italics in Table 7). Except for the latter, these are Web-specific text genres that have not been carefully edited like the newspaper articles in the TIGER treebank. As a result, they contain many typographical and grammatical mistakes, as well as tabular listings. The highest concentra-

[11] Reassuringly, this is not a computational linguistics conference, but rather an annual meeting organized by a psychotherapy journal.

	Genre	TT-SPF[a)]	TT[b)]	TnT	Stanford	SVM	UIMA
1.	*TV episode guide*	***93.89***	*90.87*	*92.79*	*92.83*	*92.78*	*89.91*
2.	news report (medicine)	**96.88**	97.12	95.92	96.16	95.68	94.26
3.	political speech	**97.52**	96.56	96.42	96.15	93.81	95.61
4.	job market news	**97.46**	93.65	96.19	96.95	95.18	95.44
5.	story (Paul of Thebes)	**95.42**	94.84	95.08	95.37	95.08	93.87
6.	exposition programme	**94.23**	92.13	92.83	92.66	93.01	90.75
7.	*online forum*	***88.01***	*79.97*	*85.56*	*84.47*	*84.51*	*84.47*
8.	report on infections	**98.25**	96.89	97.28	**98.25**	97.08	95.54
9.	*conference information*	*90.98*	*89.18*	*92.01*	*90.98*	***93.30***	*92.55*
10.	IT news (CeBIT)	93.69	92.73	92.93	94.07	94.07	**95.42**
11.	info (support programme)	97.10	98.51	98.01	**99.50**	97.01	98.02
12.	*news report (archbishop)*	*91.97*	*87.15*	*91.97*	*91.97*	***93.97***	*90.80*
13.	synopsis of cold war	96.67	94.86	96.49	95.68	95.40	**97.30**
		94.77 ±3.04	92.65 ±5.04	94.11 ±3.31	94.23 ±3.85	93.91 ±3.15	93.38 ±3.67

[a] TreeTagger with standard parameter file included in distribution
[b] TreeTagger with parameter file trained on the TIGER treebank

Table 7: Tagging accuracies for the different text genres in the DEWAC gold standard. Note that the macro-averaged means in the bottom row are different from the micro-averaged means shown in Table 4. The best result for each genre is highlighted in bold font; particularly difficult genres are printed in italics.

tion of tagging errors was found in a forum posting written entirely in lowercase by a non-native speaker, as the following excerpt shows:[12]

> ... *hallo*$_{\text{ITJ}}$ *meine*$_{\text{PPOSAT}}$ ***name***$_{\textbf{NN}}$ *ist*$_{\text{VAFIN}}$ ***nesko***$_{\textbf{ADJD}}$,$_{\text{\$,}}$ *wohne*$_{\text{VVFIN}}$ *in*$_{\text{APPR}}$ ***dubrovnik***$_{\textbf{NN}}$ *in*$_{\text{APPR}}$ ***kroatien***$_{\textbf{NN}}$.$_{\text{\$.}}$ *habe*$_{\text{VAFIN}}$ *schon*$_{\text{ADV}}$ ***stones***$_{\textbf{ADJA}}$ ***karte***$_{\textbf{NN}}$ *fur*$_{\text{XY}}$ ***olympia***$_{\textbf{ADJD}}$ ***stadion***$_{\textbf{ADJA}}$ ***konzert***$_{\textbf{NN}}$ *und*$_{\text{KON}}$ *mochte*$_{\text{VVFIN}}$ *gerne*$_{\text{ADV}}$ *auch*$_{\text{ADV}}$ *fur*$_{\text{XY}}$ ***halle***$_{\textbf{VVFIN}}$ ...

The author of this text fails to capitalize names and common nouns (highlighted in bold font) and omits the diaresis in words like *für* and *möchte* (underlined). As a result, almost every other word is not recognised by the tagger, resulting in an accuracy of only 58% for this sentence. There are also various grammatical mistakes, which would pose additional difficulties for the taggers even if there were no unknown words.

Table 7 shows that there is no single best tagger for Web texts that works equally well across all genres. Different heuristics and optimizations used by individual taggers make them particularly suitable for specific text genres. TreeTagger with its standard parameter file achieves the best accuracy for 8 out of 13 genres and works reasonably well for the remaining 5 genres. It is therefore the recommended choice for Web texts and other non-standardized genres at the current time.

6.2 Tagset granularity

Applications of Web corpora may not always require the full detail of the 54 different tags in the STTS tagset (examples include basic information mining, computational lexicography, and distributional semantic models). In such cases, a coarse-grained tagset that distinguishes, e.g., verbs from nouns and adjectives, will be sufficient. In this section, we show that mapping parts of speech to such a reduced tagset results in substantially higher tagging accuracy. Again, we use the best-performing tagger on Web texts, TreeTagger with its standard parameter file, as an example.

The TIGER treebank and the DEWAC gold standard were first tagged with the original STTS tagset (54 tags), then we mapped the output of the tagger onto a reduced tagset (14 tags for major parts of speech) before carrying out the evaluation. Tagging accuracy increases by almost 2% on TIGER, and almost 3% on the Web texts (see Table 8). There is also a drastic increase in unknown-words accuracy (by ca. 8%–14%), as many confusion pairs are now mapped to the same coarse POS tag. In particular, the most frequent errors type specific to Web texts disappear completely or are considerably reduced.

Table 9 shows separate accuracy results for each text genre in the DEWAC gold standard, using the

[12] The POS tags in this excerpt were automatically assigned by TreeTagger with its standard parameter file.

	overall	KW	UW	% unk.
TIGER treebank (TT, SPF)				
fine	95.82	96.27	79.88	2.70
coarse	97.79	97.80	93.50	2.70
TIGER treebank (TT, cross-validation)				
fine	96.90	97.62	87.89	7.40
coarse	98.28	98.50	95.60	7.40
DEWAC *gold standard (TT, SPF)*				
fine	93.71	95.42	54.30	4.15
coarse	96.51	97.81	66.50	4.15

Table 8: TreeTagger accuracy on TIGER and DEWAC for fine *vs* coarse tagset.

reduced tagset as described above. The gain in accuracy ranges from ca. 1% (for "easy" genres) up to almost 6% for particularly difficult texts. Even the online forum postings can now be tagged with an accuracy of 93.75%.

	fine tagset		coarse tagset	
#	all	unknown	all	unknown
1	*93.89*	*52.63*	*96.16*	*64.47*
2	96.88	85.71	99.04	92.85
3	97.52	58.33	98.21	58.33
4	97.46	80.00	97.97	80.00
5	95.42	68.62	97.28	72.55
6	94.23	73.91	97.90	95.65
7	*88.01*	*39.20*	*93.75*	*57.60*
8	98.25	100.00	99.42	100.00
9	*90.98*	*33.33*	*94.33*	*43.33*
10	93.69	46.42	95.79	57.14
11	97.10	33.33	99.50	100.00
12	*91.97*	*92.85*	*97.19*	*92.85*
13	96.67	27.27	97.02	36.26
	94.77	60.89	97.20	73.16
	±3.04	±24.44	±1.80	±22.01

Table 9: Comparison of tagging accuracy for fine and coarse tagset across DEWAC text genres (TT-SPF). "Difficult" genres are displayed in italics.

7 Conclusions

The goal of the study reported here was to empirically evaluate the performance of POS taggers trained on newspaper corpora in a real-world scenario, esp. when applied to less standardized text genres such as Web pages. Since there is no suitable Web reference corpus, we annotated a sample of German Web pages from the DEWAC corpus using a semi-automatic procedure. Five state-of-the art statistical taggers were trained on the TIGER treebank and evaluated on the new DEWAC gold standard.

Cross-validation on TIGER established the MaxEnt-based Stanford tagger as the best-performing tagger for German under the artificial "ideal" conditions used by most evaluation studies. Its per-word accuracy of 97.63% exceeds the published TreeTagger result of 97.54%, at the cost of much higher computational complexity (by more than a factor of 300).

When applied to Web texts, the accuracy of all taggers drops drastically, e.g. from 97.63% to 92.61% for the Stanford tagger. It is also no longer the best tagger in this scenario, being outperformed by the best HMM-based tagger TnT (92.69%). We take this result as an indication of overfitting by advanced machine-learning techniques such as MaxEnt and SVM. Surprisingly, TreeTagger achieves the lowest accuracy of all five taggers in the comparative DEWAC evaluation. Using the standard parameter file included in its distribution (which contains a heuristic lexicon extracted from a large, automatically tagged corpus), TreeTagger outperforms TnT by a margin of 1%. Its per-word accuracy of 93.71% is still not adequate for most applications, though.

A closer look at the individual texts of the DEWAC gold standard revealed that certain "easy" genres of Web pages can be tagged with state-of-the-art accuracy. Other, Web-specific genres such as online forum postings are "hard" and may result in tagging accuracies below 90%. If only a coarse-grained distinction between major parts of speech is required, a tagging accuracy of up to 96.51% can be achieved. Such a mapping to a reduced tagset is particularly beneficial for the "hard" Web genres, which can then be tagged with satisfactory accuracy (93.75% *vs* 88.01%).

We realize that making the task easier by reducing the number of tags is not an ultimate goal. The adaptation of statistical models for cross-domain tagging is currently a *hot topic* in NLP research (Finkel and Manning, 2009; Daumé III, 2009). Based on the insights from the latter and our in-depth study of POS taggers, we plan to develop more robust taggers for the Web.

Acknowledgments

This work was partially supported by German ”Federal Ministry of Economics” (BMWi) under the project Theseus (number 01MQ07019).

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: A collection of very large linguistically processed Web-crawled corpora. *Language Resources and Evaluation*. To appear.

Sabine Brants and Silvia Hansen. 2002. Developments in the tiger annotation scheme and their realization in the corpus. In *Proceedings of the Third Conference on Language Resources and Evaluation (LREC 2002)*, pages 1643–1649.

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 2002)*, Sozopol, Bulgaria.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing*, pages 224–231.

Eric Brill. 1995. Unsupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, pages 1–13.

Hal Daumé III. 2009. Bayesian multitask learning with latent hierarchies. In *Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada.

Markus Dickinson and Walt Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*, pages 107–114, Budapest, Hungary.

Jenny Rose Finkel and Christopher D. Manning. 2009. Hierarchical bayesian domain adaptation. In *Proceedings of the North American Association of Computational Linguistics (NAACL 2009)*.

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 2004)*, Lisbon, Portugal.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. HunPos – an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Posters and Demonstrations Sessions*, pages 209–212, Prague, Czech Republic.

Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the Web as corpus. *Computational Linguistics*, 29(3):333–347.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

Christof Müller, Torsten Zesch, Mark-Christoph Müller, Delphine Bernhard, Kateryna Ignatova, Iryna Gurevych, and Max Mühlhuser. 2008. Flexible UIMA components for information retrieval research. In *Proceedings of the LREC 2008 Workshop 'Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP'*, pages 24–27, May.

E. Nyberg, E. Riebling, R.C. Wang, and R: Frederking. 2008. Integrating a natural language message preprocessor with uima. In *Proceedings of the LREC 2008 Workshop 'Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP'*, pages 28–31, May.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, IMS, University of Stuttgart and SfS, University of Tübingen, August.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT Workshop*, March.

Libin Shen, Giorgio Satta, and Aravind Joshi. 2007. Guided learning for bidirectional sequence classification. In *ACL*. The Association for Computer Linguistics.

Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. 1998. A linguistically interpreted corpus of German newspaper texts. In *Proceedings of the ESSLLI Workshop on Recent Advances in Corpus Annotation*, Saarbrücken, Germany.

Pasi Tapanainen and Atro Voutilainen. 1994. Tagging accurately – Don't guess if you know. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 47–52.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

E. Ukkonen. 1995. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260.

Martin Volk and Gerold Schneider. 1998. Comparing a statistical and a rule-based tagger for German. In *Proceedings of the 4th Conference on Natural Language Processing*, KONVENS-98, pages 125–137, Bonn.