A harmonised testsuite for POS tagging of German social media data

Ines Rehbein
Leibniz ScienceCampus
IDS Mannheim

Josef Ruppenhofer
Leibniz ScienceCampus
IDS Mannheim

Victor Zimmermann
ICL
Heidelberg University

last_name@cl.uni-heidelberg.de

Abstract

We present a testsuite for POS tagging German web data. Our testsuite provides the original raw text as well as the gold tokenisations and is annotated for parts-of-speech. The testsuite includes a new dataset for German tweets, with a current size of 3,940 tokens. To increase the size of the data, we harmonised the annotations in already existing web corpora, based on the Stuttgart-Tübingen Tag Set. The current version of the corpus has an overall size of 48,344 tokens of web data, around half of it from Twitter. We also present experiments, showing how different experimental setups (training set size, additional out-of-domain training data, self-training) influence the accuracy of the taggers. All resources and models will be made publicly available to the research community.

1 Introduction

Previous work has addressed the topic of POS tagging German web data (Giesbrecht and Evert, 2009) and data from the social media (Rehbein, 2013; Prange et al., 2016).[1] However, so far no free and easy-to-use POS tagger for German web and social media data has been made available. Such a tool would fill an important need as more and more research projects are working with social media data and there is an increasing demand for robust preprocessing tools for applications in the area of sentiment analysis and opinion mining, information extraction or machine translation, to name but a few.[2] It seems fair to say that POS tagging is a crucial first step in many processing pipelines and obtaining robust results in early stages of the preprocessing will have great impact on the overall results of any tool working with non-standard language.

We address this gap by providing trained POS models for German Twitter microtext, as well as a new harmonised testsuite for training and evaluation. The new Twitter testsuite is part of a larger set of already existing web and social media data, annotated for parts-of-speech, that have been harmonised so that they can be combined for training and testing.

In addition to the tagging models and the testsuite for web and social media data, we present experiments that explore the adequacy of different tagging architectures and experimental setups for out-of-domain POS tagging where we only have a small amount of in-domain training data. We *do not* present new tools or architectures for out-of-domain POS tagging but focus on testing existing tools with regard to their accuracy, their robustness when trained on differently sized training data and their performance in a self-training setup.

The paper is structured as follows. In section 2 we review related work on POS tagging in an out-of-domain setting. In section 3 we present the new testsuite and describe the harmonisation process as well as the annotation of the new Twitter subcorpus. Section 4 describes the tagging experiments and reports our results. In section 5 we conclude and outline future work.

2 Related work

POS tagging of newspaper text has been described as a solved problem, with accuracies well over 97% (Giesbrecht and Evert, 2009). However, if we apply the same taggers to out-of-domain data, results detoriorate. Work on POS tagging of social media data has reported accuracies in the range of 85-95% accuracy (Giesbrecht and Evert, 2009; Owoputi et

[1] Also see the EmpiriST 2015 shared task on the automatic linguistic annotation of computer-mediated communication (CMC) and web corpora (Beißwenger et al., 2016).

[2] See, for example, the GermEval 2018 shared task on the Identification of Offensive Language in Twitter microtext, co-located with KONVENS 2018.

Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)
Vienna, Austria – September 19-21, 2018

al., 2013; Beißwenger et al., 2016), depending on language, text type, annotation scheme, training data size and the degree of non-canonicity in the data. Phenomena such as grammatical and spelling errors, non-standard use of upper- and lower-casing and punctuation as well as a creative use of language result in a high number of unknown word-forms that pose a problem for the tagger. This is reflected in the low accuracies for POS tagging CMC and web data.

Different approaches have been proposed to improve POS tagging results by leveraging additional unlabelled data. Søgaard (2010) uses a new variant of tri-training (Li and Zhou, 2005) to improve POS tagging on English newswire text. Other promising approaches include distributional information learned on large, unlabelled corpora in a semi- or unsupervised fashion. Owoputi et al. (2013) train Brown clusters to improve tagging performance for unknown words in English tweets. Chrupala (2011) presents a soft clustering approach based on LDA to induce soft probabilistic word classes and evaluates them in a POS tagging task. Another type of information that has been shown to improve POS tagging of out-of-domain data is distributional information from count-based context vectors (Schnabel and Schütze, 2014; Yin et al., 2015), obtained on a large unlabelled corpus.

Recent work on POS tagging has focused on neural architectures for sequence tagging. Plank et al. (2016) present a neural tagger based on bi-directional long short-term memory (bi-LSTM) networks and evaluate it on a large number of languages and different dataset sizes. They show that their model works especially well for morphologically rich languages and suggest that bi-LSTMs are not as sensitive to training size as previously assumed.

Ma and Hovy (2016) propose an end-to-end sequence labelling approach that uses information from word and character-level representations by combining a bi-directional LSTM with a CNN and adding a sequential Conditional Random Field (CRF) on top. The CNN is used for encoding character-based representations that help with unknown words, and the CRF jointly decodes the tags for the whole sentence, thus improving results over the local neural model.

Their model is similar to Lample et al. (2016) who present a bi-LSTM-CRF for Named Entity Recognition but use bi-LSTMs to learn the character embeddings.

In the paper, we test three different tagging architectures and compare their robustness to varying training data sizes as well as their performance in a self-training setup where the model has to learn in the presence of noise. As a baseline, we train a Hidden Markov Model (HMM) tagger and compare the results to the SVM-based tagger of Yin et al. (2015), a tagger specifically developed for domain adaptation that makes use of count-based distributional context features. Our third system is a state-of-the-art neural biLSTM-CRF sequence tagger that learns dense feature representations (Lample et al., 2016). Our results show that, most surprisingly, the HMM tagger yields highly competetive results on our data and that self-training is a good way of improving results in a low-resource scenario.

In the next section, we describe the annotation of the new Twitter testset and the harmonisation of the testsuite.

3 tweeDe – A new Twitter POS testsuite

Data extraction It is well known that the annotation of microtext is a challenging task, due to the brevity of the messages and the missing context information, which often results in highly ambiguous texts. To avoid such problems, we opted to extract short communication threads, which range in length from 2 up to 34 tweets. This approach allowed the annotators to see the context of each tweet and was thus crucial for resolving ambiguities in the data.

The conversations were collected in two steps. We first used an existing python tool[3] that supports the downloading of conversations by querying the Twitter API for a set of query terms and then scraping the html page on twitter.com that represents each matching conversation. However, Twitter does not embed complete json files into the html-pages and the existing crawler had some problems in fully retrieving tweet text containing certain special characters. We therefore used the output of the initial crawler only to establish the ids and the sequencing of the tweets in a conversation and then re-downloaded the full json files to be sure we had complete tweets.

The query terms we used were all German stop words, i.e. highly-frequent closed-class function words such as prepositions, articles, modal verbs,

[3] https://github.com/song9446/twitter-corpus-crawler-python

and adverbs such as *auch* 'too' or *dann* 'then'. The idea behind this was to avoid any kind of topic bias. Of the threads retrieved, we only retained those representing private communication between two or more participants. Threads consisting mainly of automatically generated tweets, advertisements, and so on were discarded after manual inspection.

Besides issues arising from brevity, further problems for annotating CMC are the creative use of language, including emoticons and acronyms (examples 1, 2, 3), non-canonical spellings (4), missing arguments (2) and the often missing or inconsistent use of punctuaction (1, 2, 3, 4). The latter causes segmentation problems like those faced in annotating spoken language where, since no punctuation is given, the annotator has to decide on the unit of analysis.

(1) hdl
have you dear
"Love you"

(2) Mache deshalb gerne mal mit
participate thus gladly MODAL PTK VERB PTK
< 3
EMOTICON
"Hence (I) like to participate once in a while < 3"

(3) Mahlzeit Arbeit Gassigang Wohnung
meal work walking the dog flat
geputzt Essen gemacht Jaaaa es ist #Freitag und
cleaned food made Yeeees it is Friday and
jetzt #hochdiehaendewochenende
now #up-the-hands-weekend

(4) Is nich wahr ich habe nur einen report bekommen
is not true I have only a report got
das sie es erhalten haben und überprüfen..
that they it received have and check..
"It's not true. I only got a report that they have received it and will check it."

For spoken language, several proposals regarding segmentation have been made, based on syntax, intonation and prosodic cues, pausing and hesitation markers (Rehbein et al., 2004; Selting et al., 2009). However, when the different levels of analysis present contradicting evidence, it is not clear how to proceed. For tweets, we have to deal with similar issues. When no (or only inconsistent use of) punctuation is present, we have to decide how to segment the tweet into units of analysis. For tweeDe, we have used the following rules to guide the segmentation.

- Hashtags and URLs at the beginning or the end of the tweet that are not integrated in the sentence are separated and form their own unit (see figure 1).
- Emoticons are treated as non-verbal comments to the text and are thus integrated in the utterance.
- Interjections (*Aaahh*), inflectives (**grins**, fillers (*ähm*) and acronyms typical for CMC (*lol, OMG*) are also not separated but considered as part of the message.

Annotation quality The STTS-POS tags that we use represent the annotations of an annotator with extensive prior experience in annotating POS tags following the STTS scheme. Annotation was not done from scratch but consists of error correcting the output of the UD processing pipeline for German (Straka and Straková, 2017), as the STTS POS annotations were created as part of a larger effort that aims to create additional data for the German Universal Dependencies (UD) treebank. Within the larger project, two annotators provided syntactic dependency labels, which were subsequently adjudicated. The STTS POS annotator also corrected the UD POS tags. The adjudicated syntactic dependency relations were used for consistency checks between the dependency labels and the STTS tags. Additional consistency checks verified the compatibility of UD POS labels and STTS POS labels. All incompatibilities were manually inspected and resolved.

The current version of the Twitter testsuite used in the experiments presented here includes 3,940 tokens. We are also working on increasing the testsuite, with a planned final size of 500 sentence-like units[4] for testing and 250 units for development.

3.1 Data harmonisation

To obtain a dataset for German web and social media data, we combine our new testsuite with already existing datasets and harmonise the data so that all subsets use the same annotation scheme and are segmented and tokenised according to the same rules. In addition to tweeDe, the harmonised data contains the training and test sets from the EmpiriST 2015 shared task (Beißwenger et al., 2016) (23,223 tokens) as well as the Twitter dataset used in Rehbein (2013) (21,181 tokens). We refer to these datasets as EmpiriST and Tw2013.

The EmpiriST data includes different types of computer-mediated communication (CMC)[5] as

[4] By that we mean messages with at least 3 tokens that do not include at-mentions, hashtags or URLs only.

[5] Private chat, professional chat, tweets sent in an academic context, chat from Wikipedia talk pages, WhatsApp messages and blog comments; see Beißwenger et al. (2016)

name	# tok	train/dev/test split	# vocab	TTR	OOV_{same}	OOV_{TiGer}	% ambig. tags	#tags
TiGer	360,200	323,343/-/36,857	48,176	.486	10%	10%	16.2	50
$EmpiriST_{CMC}$	10,505	5,176/-/5,329	3,549	.494	32%	27%	16.2	52
$EmpiriST_{web}$	12,718	5,029/-/7,689	4,281	.498	41%	19%	16.4	49
Tw2013	21,181	6,305/7,342/7,534	8,006	.509	39%	30%	16.8	52
tweeDe	3,940	-/-/3,940	1,509	.507	-	16%	17.0	51

Table 1: Corpus statistics for the different subsets in the testsuite and for a subcorpus from TiGer (# tok: no. tokens in the whole subcorpus and in the train/dev/test sets; vocabulary size; TTR: type-token-ratio averaged over 1000 equal-sized samples of 3000 tokens; OOV: % out-of-vocabulary words with regard to the training set from the same subcorpus (OOV_{same}) and TiGer training data (OOV_{TiGer}); ratio of ambiguous POS tags; #tags: no. different tags in the testsets).

well as data scrawled from the web. The Tw2013 data of Rehbein (2013) contains randomly selected tweets downloaded between July 2012 and February 2013. Table 1 shows some statistics for the harmonised testsuite.

The mapping used for harmonising the different annotation schemes is from fine to coarse, thus omitting information that has been present in the original annotations. The goal of our endeavour, however, is to provide annotations that are reasonably frequent even in smaller datasets and thus are easier to learn for automatic systems.

Table 2 shows the modifications applied to the data. All tags are taken from the STTS (Schiller et al., 1995), with the addition of HASH for hashtags[6] and EMO for emoticons. We also removed the contracted POS tags (e.g. PPER_PPER, VVFIN_PPER, KOUS_PPER etc.) used in the EmpiriST and Tw2013 annotations as we think that contractions are a problem that should be handled during tokenisation (see figure 1).

For all instances in the testsuite, we provide the data in CoNLLU format, including the original raw text as well as the tokenised version of the data. This will allow researchers to train their own tokeniser or to use existing tools such as the UD-Pipeline that accepts CoNLLU as input format.

Following the UD guidelines for tokenisation, we also split contracted prepositions and determiners (e.g. *im* → *in dem* "in the", *fürn* → *für den* "for the"). This provides a consistent treatment for grammaticalised contractions such as *zum* (e.g. *zum Beispiel* "for example" where the split form *zu dem Beispiel* "for the example" is highly non-canonical) and for less grammaticalised and highly informal contractions such as *aufn* "on the". However, the format preserves the original information which can be easily extracted to replace the split version, if desired.

tag	*EmpiriST*	*Tw2013*	*mod.*
URLs	URL	URL	NE
hashtags	HST	HASHTAG	HASH
at-mentions	ADR	ADDRESS	NE
emoticons	EMOASC, EMOIMG	EMO EMO	EMO EMO
inflectives,	AKW	COMMENT	VROOT
acronyms. etc.		COMMENT	ITJ
e-mail addr.	EML	–	NE
intensifier	PTKIFG	ADV	ADV
modal particle	PTKMA	ADV	ADV
connectives + V2	DM	KON	KON
onomatopoeia	ONO	PTKONO	ITJ
filler	–	PTKFILL	ITJ
backchannel	–	PTKREZ	ITJ
tag question	–	PTKQU	ITJ
placeholder	–	PTKPH	
unfinished word	–	XYB	XY
uninterpretable	–	XYU	XY

Table 2: Original tags and modified tags after harmonising the datasets.

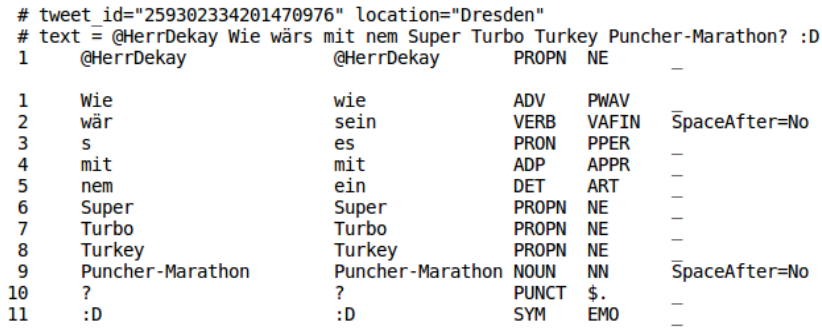
4 Experimental setup

We now compare the performance of different POS taggers on the harmonised testsuite, investigating the impact of text type and data size as well as the ability of the models to learn in the presence of noise.

As additional training data to our web and social media data, we use 360,200 tokens of newspaper text from the TiGer treebank (Brants et al., 2002), annotated for POS. We also make use of a large collection of unlabelled tweets (9.6GB) to train skip-gram word embeddings (Mikolov et al., 2013a; Mikolov et al., 2013b) to initialise the feature weights for the neural sequence tagger.[7] A subset of the same data has also been used in the self-training experiments described below.

[6] We consider URLs and at-mentions to be named entities while hashtags can be associated with many different POS tags as well as with multi-word expressions or whole phrases, see example (3) above.

[7] We used the Gensim implementation of the skip-gram model, trained with 100 dimensions, a window size of 7 and a min. word count of 50: `https://radimrehurek.com/gensim/`



```
# tweet_id="259302334201470976" location="Dresden"
# text = @HerrDekay Wie wärs mit nem Super Turbo Turkey Puncher-Marathon? :D
1     @HerrDekay              @HerrDekay       PROPN  NE       _

1     Wie                     wie              ADV    PWAV     _
2     wär                     sein             VERB   VAFIN    SpaceAfter=No
3     s                       es               PRON   PPER     _
4     mit                     mit              ADP    APPR     _
5     nem                     ein              DET    ART      _
6     Super                   Super            PROPN  NE       _
7     Turbo                   Turbo            PROPN  NE       _
8     Turkey                  Turkey           PROPN  NE       _
9     Puncher-Marathon        Puncher-Marathon NOUN   NN       SpaceAfter=No
10    ?                       ?                PUNCT  $.       _
11    :D                      :D               SYM    EMO      _
```

Figure 1: Data format, segmentation and tokenisation of the data in CoNLLU for one tweet. 2-3 shows a contracted token that has been split into 2 separate tokens. The "SpaceAfter" attribute in col. 5 indicates that there is no space after token 9.

Hidden Markov Model tagging We start with the Hunpos tagger (Halácsy et al., 2007), an open-source HMM tagger. The Hunpos model trained on TiGer yields high results on the TiGer testset (97.0% acc.; table 3) . When applying the TiGer-trained tagger to the different subcorpora in the test-suite, we see a substantial decrease in performance (table 3). On the EmpiriST web subcorpus the results are still reasonably high with 90.8% while on the CMC data the accuracy decreases to 73.7%. Results for the two Twitter testsets are somewhere in between with 79.8% for Tw2013 and 84.9% for tweeDe.

The variation in results for the different subcorpora shows that either our datasets are not big enough to provide representative results or that we have to account for a considerable amount of variation even for data from the same source, such as Twitter microtext. The higher accuracy for tweeDe for the TiGer-trained tagger might reflect the lower amount of OOV words with respect to the TiGer training data (OOV: 16%; table 1) as compared to the OOV of 30% in the Tw2013 dataset.

Table 3 (lines 2-4) shows results for training and testing Hunpos on the individual subcorpora. Given the small size of the training data, the low results are not surprising. As expected, we can substantially increase results when combining the training data from the different subcorpora. Training on a combination of all available datasets yields best results for nearly all subcorpora. We can see that even small amounts of in-domain training data can substantially boost results, leading to accuracies for all three social media testsets of close to 87.%.

training	TiGer	Web	CMC	Tw2013	tweeDe
1 TiGer	**97.0**	90.8	73.7	79.8	84.9
2 Web	79.7	85.9	68.5	69.4	75.5
3 CMC	75.8	80.9	83.0	69.9	77.5
4 Tw2013	80.0	82.5	72.9	81.3	79.4
5 1+2+3	**97.0**	**93.0**	86.5	86.5	86.7
6 1+4	**97.0**	91.8	78.2	86.2	86.6
7 1+2+3+4	**97.0**	92.9	**86.6**	**86.9**	**86.8**

Table 3: Baseline results for HunPos trained on different subcorpora and dataset combinations (1+2+3: TiGer + Web + CMC, etc.)

Count-based distributional context vectors Next, we test the Online-FLORS tagger (Yin et al., 2015) which has been explicitly designed for domain adaptation. Online-FLORS is based on the FLORS tagger (Schnabel and Schütze, 2014) but replaces its batch learning strategy with an online learning approach. The tagger uses an SVM classifier trained on four feature types: shape and suffix features and distributional features capturing the left and the right context of a word. The distributional feature vectors are created based on the word counts in a corpus of unlabelled in-domain data. In our experiments, we use a set of 1 mio. German tweets that have been tokenised before training. Besides tokenisation, no further preprocssing has been applied.

Table 4 shows that the tagger, based on the distributional information from the raw Twitter data, is able to improve results substantially. The largest improvements can be observed for the web-trained tagger applied to the CMC test set (table 4, line 2: +17.3% as compared to Hunpos (68.5%)).

As before, we obtain best results when training on a combination of all subcorpora. The boost in results, however, is not as impressive as before when training on individual subcorpora, but we are still able to increase accuracies for the CMC and Twitter subcorpora in the range of 1 to 2.3 percentage points. The tagger also outperforms the HMM tag-

training	TiGer	Web	CMC	Tw2013	tweeDe
1 TiGer	97.3	92.7	78.6	80.4	87.0
2 Web	86.2	89.1	85.8	75.3	81.5
3 CMC	83.4	75.6	86.7	78.5	82.3
4 Tw2013	85.0	85.7	79.1	87.1	83.5
5 1+2+3	**97.3**	**93.2**	**87.7**	89.1	**88.6**
6 1+4	97.2	93.0	80.8	88.8	88.2
7 1+2+3+4	97.2	**93.2**	87.5	**89.4**	**88.6**

Table 4: Baseline results for Online-Flors trained on different subcorpora and dataset combinations (1+2+3: TiGer + Web + CMC, etc.)

ger on the user-generated testsets by 0.9 to 2.5%, showing that the domain adaptation strategy of using count-based distributional information is highly successful. These results show again the potential of distributional methods for out-of-domain POS tagging. In the EmpiriST shared task the winning system also used a distributional approach to overcome data sparseness.

Prediction-based context information Next, we would like to see how a neural tagger initialised with prediction-based pretrained word embeddings compares to the SVM tagger that draws its auxiliary information from count-based context vectors.

Neural methods have made tremenduous progress in recent years and many implementations for sequence tagging are available. For our experiments, we chose the implementation of Reimers and Gurevych (2017) that provides the functionality of two recent state-of-the-art sequence tagging models (Ma and Hovy, 2016; Lample et al., 2016). We first train a bidirectional LSTM with word embeddings and a CRF on top. The model is similar to the one of Plank et al. (2016) which is also based on biLSTMs but has an additional CRF on top that jointly decodes the tags for the whole sentence. The model was trained for 50 epochs (with early stopping) using the default parameter settings (dimensions: 100, dropout: 0.25, batch size: 32, optimizer: Adam).

We test two different versions of pretrained word2vec embeddings, i) the ones used in Reimers et al. (2014), trained on 116 mio. German sentences[8] and ii) the skip-gram embeddings trained on around 108 Mio. microtexts from Twitter.

As Reimers and Gurevych (2017) have shown the importance of reporting score distributions for non-deterministic methods instead of single performance scores, in table 5 and in all subsequent result tables we report averaged scores over 5 runs for each training-test combination. In each run, we chose the model that yields best results on the Tw2013 development set and report the results for this model on the individual test sets.

training	TiGer	Web	CMC	Tw13	tweeDe
	embeddings of (Reimers et al., 2014)				
1 TiGer	97.7	93.1	78.4	78.5	86.4
	Twitter embeddings				
1 TiGer	96.2	93.0	78.1	78.5	83.0
	embeddings of (Reimers et al., 2014)				
2a Web	83.2	84.3	68.7	70.8	73.5
	Twitter embeddings				
2b Web	85.1	87.5	71.7	72.9	78.9
	embeddings of (Reimers et al., 2014)				
3a CMC	80.8	81.3	80.4	72.8	75.9
	Twitter embeddings				
3b CMC	83.6	85.0	83.6	74.1	79.7
	embeddings of (Reimers et al., 2014)				
4 Tw2013	82.4	81.2	75.6	83.5	78.8
	Twitter embeddings				
4 Tw2013	84.6	86.1	78.4	86.2	81.4
	embeddings of (Reimers et al., 2014)				
7 1+2+3+4	97.4	94.2	87.0	89.6	88.3
	Twitter embeddings				
7 1+2+3+4	97.3	94.3	87.2	90.2	88.5

Table 5: Results for the biLSTM-CRF sequence tagger, trained on different subcorpora and dataset combinations (all results averaged over 5 runs with default parameters).

Table 5 (lines 2a,b and 3a,b) shows that, not surprisingly, the word embeddings trained on Twitter are more suitable for POS tagging CMC and social media data. For tweeDe, this results in an improvement of more than 5% for the model trained on the web data, and of nearly 4% for the CMC-trained model.

However, while most results for the individual test sets are higher than the ones obtained by HunPos (lines 1-4), in most settings the biLSTM-CRF tagger is outperformed by Online-Flors. When training the tagger on a larger set where we combine all subcorpora (table 5, line 7), the results are more or less in the same range. This shows that the two approaches, using count-based and prediction based feature vectors to overcome data sparsity, are both highly successful.

While our setting is certainly not controlled enough to contribute to the discussion on *count or predict* (Baroni et al., 2014; Levy and Goldberg, 2014; Riedl and Biemann, 2017), we find it interesting that despite having access to prediction-

[8]The embeddings are available from the website: https://www.informatik.tu-darmstadt.de/ukp/research_6/ukp_in_challenges/germeval_2014/index.en.jsp.

training	TiGer	Web	CMC	Tw13	tweeDe
1 TiGer	97.6	94.2	79.4	79.4	86.7
2 Web	86.3	88.3	72.2	73.4	79.4
3 CMC	84.7	86.5	86.7	78.2	81.4
4 Tw2013	86.1	87.2	80.6	88.7	83.4
7 1+2+3+4	97.8	95.1	90.1	92.5	89.2

Table 6: Results for the biLSTM-char-CRF sequence tagger, trained on different subcorpora and dataset combinations (all results averaged over 5 runs using pretrained Twitter embeddings and default parameters).

training	TiGer	Web	CMC	Tw13	tweeDe
1 TiGer	97.6	94.2	79.4	79.4	86.7
2 Web	86.8	88.6	72.7	74.2	80.0
3 CMC	84.3	86.3	86.2	78.9	81.3
4 Tw2013	86.8	87.6	81.4	89.1	84.0
7 1+2+3+4	97.9	95.1	90.5	92.6	89.4

Table 7: Results for the biLSTM-char-CRF sequence tagger, trained on different subcorpora and dataset combinations (all results averaged over 5 runs using pretrained Twitter embeddings, dimensions: 70, dropout: 0.45, minibatch size: 32).

based distributional information extracted from a much larger ressource (>108 mio. tweets), the biLSTM-CRF sequence model fails to outperform the SVM-based tagger that utilises count-based context information obtained from a much smaller corpus with only 1 mio. tweets. Unfortunately, the Online-Flors requires a lot of memory which makes it unfeasible to test whether training on a larger unlabelled corpus might further increase results.

Character embeddings Character-based representations have been shown to improve results for many different NLP tasks such as POS tagging, dependency parsing, MT or sentiment analysis (Plank et al., 2016; Ling et al., 2015; Costa-jussà and Fonollosa, 2016; Jebbara and Cimiano, 2017). Their potential to generalise to rare and out-of-vocabulary words makes them especially useful for dealing with morphologically rich languages (Ballesteros et al., 2015) or out-of-domain settings (Dhingra et al., 2016).

To test the potential of character-based embeddings for POS tagging of German tweets, we train a biLSTM model with word and character embeddings, following the approach of Lample et al. (2016) to learn the character representations based on LSTMs[9] (table 6).

As expected, the character-based representations yield further improvements especially for the user-generated content where we encounter many out-of-vocabulary words. When applying the TiGer-trained model initialised with pre-trained Twitter word embeddings to the tweeDe testset, we observe an increase in accuracy of over 3%. For the model trained on a combination of all four datasets, we also see improvements in the range of 0.5% (TiGer) up to 2.9% (CMC).

Parameter optimisation Parameter settings can have a crucial impact on the performance of neural models. To explore the impact of different parameter settings on tagging accuracy, we systematically vary the number of hidden units in the range of $\{50, 70, 100, 125, 150, 200\}$ and test the following dropout rates $\{0.25, 0.30, 0.35, 0.40, 0.45, 0.50\}$[10]. We also tried different minibatch sizes of $\{16, 24, 32, 48, 64\}$. As before, all results are averaged over 5 runs, and for each run we train for 50 iterations using early stopping and select the model that yields best results on the Tw2013 development set.

Table 7 shows results for the best parameter combination (dimemsion size: 70, dropout: 0.45, minibatch size: 32). For the model trained on the combined datasets (line 7), we only observe insignificant improvements. When training the model on the much smaller individual subcorpora, however, we see that the parameters are more important and yield further improvements of up to 0.8% (line 4, training data: Tw2013, testset: CMC).

Self-training In our final experiment, we test the potential of self-training for improving results on the social media data. For this, we use an ensemble of POS taggers trained on the combination of all datasets, to predict the tags for additional unlabelled tweets from our Twitter corpus. We use our best models for the HunPos, Online Flors and biLSTM-char-CRF taggers, all trained on the combined training data from TiGer and all four subcorpora in the testsuite. To give more importance to the in-domain training data, we oversample these

[9]Reimers and Gurevych (2017) only observed insignificant differences in POS tagging results for the LSTM-based approach of Lample et al. (2016) and the one of Ma and Hovy (2016) who use CNNs for learning character-based representations.

[10]We use *variational dropout* (Gal and Ghahramani, 2016) where the same dropout mask is applied to all time steps in the same sentence and dropout is also applied to the recurrent units.

training	TiGer	Web	CMC	Tw13	tweeDe
	w/o self-training				
HunPos	97.0	92.9	86.6	86.9	86.8
O-Flors	97.2	93.2	87.5	89.4	88.6
biLSTM-C	97.9	95.1	90.5	92.6	89.4
	with 1 iteration of self-training				
HunPos	96.8	93.6	88.4	89.7	89.2
Onl. Flors	97.2	93.9	88.9	90.5	89.6
biLSTM-C	97.3	95.0	90.8	92.8	90.3
	with 2 iterations of self-training				
HunPos	96.8	94.4	88.5	90.2	89.7
Onl. Flors	97.2	94.0	88.7	91.0	90.3
biLSTM-C	97.2	95.0	90.6	92.9	90.5
	with 3 iterations of self-training				
HunPos	96.8	94.0	89.0	90.8	90.2
biLSTM-C	97.2	94.7	90.3	92.9	90.6

Table 8: Results for the self-training setup (biLSTM-char-CRF results averaged over 5 runs).

instances by a factor of 10. Then we split the additional unlabelled data into three sets of around 100,000 tweets each and use the trained taggers to assign a score to each tagged message in the first set as follows.

For each tweet, we compare the tags predicted by the three tagging models and count each token where the taggers disagree as an error. For each such token, we increment the counter and after processing the whole sentence we normalise by sentence length. Then we select all sentences with a score of ≤ 0.1.

In addition, we use a similar counter for punctuation marks (here the counter is incremented for each token where at least one tagger predicted a punctuation mark) and remove all sentences with a punctuation score > 0.6.

The motivation for this is that we do not want to keep incorrectly tokenised sentences that mostly consist of punctuations. We also decided against excluding all sentences with an error score > 0 as this might lead to a bias towards very short and easy-to-tag messages that won't provide new information for learning.

After selecting the sentences for self-training, we use a majority vote to assign a tag to each token. In case of ties, we randomly select one tag. Then we add the selected, automatically tagged tweets to our training data and retrain the taggers.[11]

Table 8 shows the results for self-training. Not surprisingly, after one iteration of self-training the results for TiGer slightly decrease for two of the three taggers as the automatically tagged tweets do not provide new information for tagging newspaper text. For the user-generated content, however, we see a nice boost in results. Overall, it seems as if the HMM tagger can make the most of the additional noisy data, now obtaining results that are only slightly lower than the ones for the other taggers. This is surprising, given that the HMM tagger has no access to additional distributional information as provided by word embeddings or count-based context vectors.

For the Tw13 dataset, the neural model yields much higher results than the other two taggers. This is because we used the development set from the Tw13 data to select the best performing model, thus introducing a bias towards the Tw13 dataset. In addition, this shows that we should not think of Twitter microtext as a homogenous text genre but should keep in mind that private conversation (as included in tweeDe) is very different from other Twitter messages such as news tweets, ads, announcements etc.

To see whether more iterations of self-training can further increase results, we add another iteration of self-training where we used the models trained on the former self-trained corpus to predict the tags for the next set of unlabelled tweets and combined this data with our labelled training set. Table 8 shows that the higher accuracy for the unlabelled messages results in a small increase in results after the second iteration of self-training.

In a final experiment, we use the tagger models from the second iteration and predict tags for the first and third unlabelled self-training dataset (200,000 tweets). Again, we filter the tweets based on the predictions of the three taggers and train the HunPos tagger and the neural classifier on the combination of the labelled and unlabelled dataset. We did not include Online-Flors because of its huge memory requirements. The results for the final models show further improvements for HunPos while the results for the neural tagger are more or less in the same range as after the second iteration.

Error analysis We now want to find out what the main sources of error are and whether the different models perform equally well on different parts-of-speech. We thus evaluate the three taggers on the combined data in the testsuite (not including the TiGer subcorpus). Table 9 shows the tags most often confused by the three taggers. Error type `NN/NE` in the first row refers to an error where

[11]The Online Flors tagger already provides an option '-labeledData' that we used to include the additional self-training data. For the other taggers, we simply add the new data to the training sets and retrain the models.

Error type	HunPos	Flors	biLSTM
NN / NE	387	356	318
HASH / NE	136	131	48
$(/ $.	128	112	112
ADJD / ADV	85	81	87
VVFIN / VVINF	72	58	58
FM / NE	69	50	39
NN / ADJA	61	52	47
ITJ / NE	55	44	33
HASH / NN	40	40	32

Table 9: Most frequent errors made by the different taggers (ADJA: attributive adjective, ADJD: predicative/adverbial adjective, ADV: adverb, FM: foreign word, HASH: hashtag, ITJ: interjection, NN: noun, NE: proper name, VVFIN/VVINF: (in)finite full verb, $(/$.: punctuation).

either the gold tag was a noun (NN) and the tagger predicted a proper name (NE), or the predicted tag was a noun and the gold tag a proper name.

We can see that all taggers struggle with the same POS tags, with one notable exception. The character-based model allows the biLSTM tagger to identify hash tags with a much higher accuracy while the other two taggers have more difficulties with this particular tag.[12]

The high number of punctuation errors is partly due to the annotation procedure in the EmpiriST data where colons are either tagged as sentence-final punctuation ($.) or as sentence-medial ($() while according to the STTS guidelines all colons are considered as sentence-final.

Comparison with the EmpiriST shared task To put our work in context, we would like to compare our results to the ones from the EmpiriST shared task. Unfortunately, the different annotation schemes make a direct comparison impossible. Results for the shared task, however, were not only reported for the extended annotations in the EmpiriST data but also for a mapping from the EmpiriST annotations to STTS tags. We thus implemented the same mappings to get an idea how our taggers perform in comparison to the best shared task systems.

Table 10 shows results for the the best and second best system from the EmpiriST shared task. Our best results for HunPos are a little below the ones for the shared task winner (Prange et al., 2016)

[12]The Online-Flors tagger makes use of shape features that can be optimised to provide a better handling of hashtags. We did not do this in our experiments as the source code release only contains the class files but not the java source code.

	training	**Web**	**CMC**
	HunPos	93.7	89.9
	biLSTM-char-CRF	**94.8**	**90.5**
	EmpiriST shared task		
1.	UDS distrib.	94.6	90.3
2.	LTL-UDE	93.1	88.8

Table 10: Comparison with the best and second best systems from the EmpiriST 2016 shared task (mapping to STTS 1.0)

but better than the system ranked second (Horsmann and Zesch, 2016) while the neural model after self-training[13] shows results in the same range as the shared task winner.

5 Conclusions

We presented a new testsuite for POS tagging of German web and social media data, with a large portion of microtext from Twitter. The harmonised testsuite has a size of over 48,000 tokens and includes the original raw text and the tokenised version, annotated for POS. The data will be made available to the research community and will hopefully provide the basis for future work on tagger adaptation to social media text, thus resulting in more accurate and robust tools.

We provided baselines for three different tagging models on the data, a conventional HMM tagger, an SVM classifier augmented with count-based distributional features, and a neural biLSTM-char-CRF tagger with pretrained word embeddings. Our experiments showed that i) for training on very small datasets, additional distributional information extracted from unlabelled data is crucial; ii) combining data from different sources is beneficial in most settings; iii) even small amounts of annotated in-domain data can substantially increase results over training on out-of-domain data only; and iv) we can easily obtain further improvements using self-training.

Our most surprising finding is that, given enough training data, the HMM tagger yields accuracies only slightly below the ones for the other two models that leverage additional distributional knowledge learned from large unlabelled data. The HMM tagger also seemed to be the tagger that was best suited to learn in the presence of noise, as shown in our self-training setup. All data and trained models are available from the first author's website.

[13]The pretrained embeddings used in the evaluation were trained on lower-cased tweets with a min. word count of 10. Other parameters are as described above.

Acknowledgments

This research has been conducted within the Leibniz Science Campus "Empirical Linguistics and Computational Modeling", funded by the Leibniz Association under grant no. SAS-2015-IDS-LWC and by the Ministry of Science, Research, and Art (MWK) of the state of Baden-Württemberg.

References

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, pages 349–359.

Marco Baroni, Georgiana Dinu, and Germá Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *The 52nd Annual Meeting of the Association for Computational Linguistics*, pages 238–247.

Michael Beißwenger, Sabine Bartsch, Stefan Evert, and Kay-Michael Würzner. 2016. EmpiriST 2015: A Shared Task on the Automatic Linguistic Annotation of Computer-Mediated Communication and Web Corpora. In *Proceedings of the 10th Web as Corpus Workshop*, pages 44–56. Association for Computational Linguistics, August.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, TLT'02.

Grzegorz Chrupala. 2011. Efficient induction of probabilistic word classes with LDA. In *IJCNLP*, pages 363–372.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *The 54th Annual Meeting of the Association for Computational Linguistics*, pages 357–361.

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. 2016. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 269–274. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 1027–1035.

Eugenie Giesbrecht and Stefan Evert. 2009. Is part-of-speech tagging a solved task? an evaluation of pos taggers for the german web as corpus. In I. Alegria, I. Leturia, and S. Sharoff, editors, *Proceedings of the 5th Web as Corpus Workshop (WAC5)*, San Sebastian, Spain.

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, Prague, Czech Republic.

Tobias Horsmann and Torsten Zesch. 2016. UDE @ EmpiriST 2015: Tokenization and PoS tagging of social media text. In *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 120–126.

Soufian Jebbara and Philipp Cimiano. 2017. Improving opinion-target extraction with character-level word embeddings. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 159–167.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Ming Li and Zhi-Hua Zhou. 2005. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 11(17):1529–1541.

Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer amd Alan W Black, and Isabel Trancoso. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, pages 1520–1530.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL'16, pages 1064–1074.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL*, pages 380–390. The Association for Computational Linguistics.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL'16, pages 412–418, Berlin, Germany.

Jakob Prange, Andrea Horbach, and Stefan Thater. 2016. UdS-(retrain—distributional—surface): Improving POS tagging for OOV words in German CMC and web data. In *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 97–105.

Jochen Rehbein, Thomas Schmidt, Bernd Meyer, Franziska Watzke, and Annette Herkenrath. 2004. *Handbuch für das computergestützte Transkribieren nach HIAT*. Sonderforschungsbereich 538.

Ines Rehbein. 2013. Fine-grained pos tagging of german tweets. In *International Conference of the German Society for Computational Linguistics and Language Technology (GSCL)*.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Empirical Methods in Natural Language Processing*, EMNLP, pages 338–348.

Nils Reimers, Judith Eckle-Kohler, Carsten Schnober, Jungi Kim, and Iryna Gurevych. 2014. Germeval-2014: Nested named entity recognition with neural networks. In *Workshop on GermEval 2014 Named Entity Recognition Shared Task, KONVENS*.

Martin Riedl and Christian Biemann. 2017. There's no 'count or predict' but task-based selection for distributional models. In *12th International Conference on Computational Semantics*, IWCS'17.

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

Tobias Schnabel and Hinrich Schütze. 2014. FLORS: Fast and simple domain adaptation for part-of-speech tagging. *Transactions of the Association for Computational Linguistics (TACL)*, 2:15–26.

Margret Selting, Peter Auer, Dagmar Barth-Weingarten, Jörg R. Bergmann, Pia Bergmann, Karin Birkner, Elizabeth Couper-Kuhlen, Arnulf Deppermann, Peter Gilles, Susanne Günthner, et al. 2009. Gesprächsanalytisches Transkriptionssystem 2 (GAT 2). *Gesprächsforschung: Online-Zeitschrift zur verbalen Interaktion*.

Anders Søgaard. 2010. Simple semi-supervised training of part-of-speech taggers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL'10, pages 205–208. The Association for Computer Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wenpeng Yin, Tobias Schnabel, and Hinrich Schütze. 2015. Online updating of word representations for part-of-speech tagging. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, pages 1329–1334.