LTL-UDE @ EmpiriST 2015: Tokenization and PoS Tagging of Social Media Text

Tobias Horsmann **Torsten Zesch**
Language Technology Lab
Department of Computer Science and Applied Cognitive Science
University of Duisburg-Essen, Germany
{tobias.horsmann,torsten.zesch}@uni-due.de

Abstract

We present a detailed description of our submission to the EmpiriST shared task 2015 for tokenization and part-of-speech tagging of German social media text. As relatively little training data is provided, neither tokenization nor PoS tagging can be learned from the data alone. For tokenization, our system uses regular expressions for general cases and word lists for exceptions. For PoS tagging, adding unsupervised knowledge beyond the available training data is the most important factor for reaching acceptable tagging accuracy. A learning curve experiment shows furthermore that more in-domain training data is very likely to further increase accuracy.

1 Introduction

Tokenization and part-of-speech (PoS) tagging are two fundamental NLP tasks. Tokenization aims at detecting word and sentence boundaries in text while PoS tagging uses the recognized words and assigns each word its syntactical category. Both tasks are especially challenging when applied on noisy social media texts (Eisenstein, 2013).

The main challenge when tokenizing social media text is the ambiguity of punctuation characters which occurs more frequently than in other domains. A major source of ambiguity are emoticons that show a surprising degree of complexity ranging from two-character emoticons such as *:)* to n-character emoticons such as *\(*.*#)*. Additionally challenges are introduced by missing whitespace characters and the use of non-standard abbreviations such as in *[...] aus meiner (Doz.)Sicht.:) [...]*.

For PoS tagging, the main source of error are the frequently occurring unknown word forms that are spelling variations of words found in the dictionary. Those spelling variations are usually not contained in the (newswire) training data of the model which leads to a strong decline in accuracy on social media data (Ritter et al., 2011; Eisenstein, 2013).

There has been little work for German social media processing, the EmpiriST (Beißwenger et al., 2016) provides for both tasks two data sets composing of dialogical and monological text of the social media domain to help the development of robust tools for German. The results of our approaches for tokenization and PoS tagging are reported under the name *LTL-UDE* in the EmpiriST rankings.

2 Tokenization

While tokenization usually comprises of two subtasks (sentence boundary detection and token boundary detection), in the EmpiriST shared task, the sentence boundaries are already given and only the token boundaries should be detected.

2.1 Task Analysis

A main challenge in this task lies in dealing with missing whitespace characters, Table 1 shows a few examples with their correct tokenization. In case (1), it is difficult to determine that in the character sequence '?"<-' the arrow symbol form a semantic unit that should not be split. This problem occurs in various forms such as in (2) where a dot indicates an abbreviation and a following word appear as single token, case (3) shows how numbers and following punctuation marks form a token and cannot just be separated.

While (1) is a case which might be solved by regular expressions, (2) requires to know that the first word is an abbreviation to which the dot belongs. An additional challenge comes from the

Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task, pages 120–126,
Berlin, Germany, August 7-12, 2016.

	(1)	(2)	(3)
Raw	*pdf?"<-Wenn*	*schriftl.Äquivalent*	*v.14.4*
Tokenized	pdf␣?␣"␣<-␣Wenn	schriftl.␣Äquivalent	v.␣14.␣4

Table 1: Examples of missing whitespace characters and their correctly tokenized form

tokenization rules defined in the EmpiriST guidelines. For example the version number *v.14.4* in (3) should be tokenized as *v.␣14.␣4* even if it is actually one entity.

2.2 Implementation

Our tokenizer performs three steps: In the first step, we split the input text into units at every whitespace character. In the second step, we use regular expressions to refine the splitting by separating alpha-numerical text segments from punctuation characters. This will also erroneously split up smilies and other character sequences. Thus, in the third step, we re-assemble sequences of punctuation characters which have been separated in the previous step. This mainly serves to restore smilies but also other symbols such as arrows and alike. We examined the training data to find the most common combinations of those character sequences and merge them to a single token when we encounter them. Furthermore, we use word lists to merge abbreviations with their following dot character. The list of abbreviations are obtained from the Tüba-DZ corpus (Telljohann et al., 2004), the German Web1T uni-gram corpus (Brants and Franz, 2006), and lists we manually obtained from Wikipedia.

Baseline Systems We compare our approach to three reference systems: a plain whitespace tokenization (i.e. the first step of our approach), tokenization with the Break-Iterator-Segmenter (BreakIter) as implemented in the NLP DKPro Core framework (Eckart de Castilho and Gurevych, 2014), and a specialized social media tokenizer from the ArkTools suite (Gimpel et al., 2011). Whitespace tokenization and BreakIter are expected to perform poorly as neither tool is designed for processing social media text. The ArkTools tokenizer is tailored to English Twitter messages which are quite similar to the EmpiriST dataset, but will obviously not capture phenomena that are specific for German.

2.3 Results & Discussion

In Table 2, we show the results of applying our methods and baseline systems to the provided training and test data. The CMC data set is harder to tokenize than the Web data. Our approach performed well on the training data set but fails to generalize to unseen data. Of our baselines systems, ArkTools is the only competitive one, which is not surprising as it aims at tokenizing tweets which are a subdomain of the provided data.

Challenging cases for our approach are situations when more than two tokens have to be separated because several whitespace characters are missing or punctuation marks belonging to abbreviations are involved. Table 3 shows examples for a few selected error cases. Example (1) shows a case of a dot terminated abbreviation which is not contained in our word lists. Example (2) shows an issue when more than one whitespace character is missing. We experimented with splitting camel case expressions but found on the training data that it does more harm than good and decided not to implement such a rule. In example (3) an abbreviation is involved which is based on two words shortened to a single letter each followed by a dot character. This abbreviation had to be split up into two tokens consisting of a letter and a dot in order to conform to the tokenization guidelines.

3 Part-of-Speech Tagging

Tagging social media text with off-the-shelve PoS taggers leads to a huge drop in accuracy compared to tagging newswire text (Ritter et al., 2011; Horsmann et al., 2015). The main cause for this drop is the high rate of out-of-vocabulary words, which are mainly caused by orthographical variations of known words (Eisenstein, 2013).

3.1 Shared Task Data

The EmpiriST training dataset contains about 10k tokens of PoS annotated German social media text (the test data contains about 13k tokens). The dataset is annotated with an extended version of the STTS tagset which adds 18 new PoS tags to account for German social media phenomena

	Method	CMC P	CMC R	CMC F_1	Web P	Web R	Web F_1	∅ F_1
Train data	Whitespace	81.7	99.9	89.8	84.4	100	91.5	90.7
	BreakIter	99.4	90.2	94.5	99.7	98.3	99.0	96.8
	ArkTools	98.7	98.7	98.7	98.2	99.2	98.7	98.7
	LTL-UDE	99.7	99.7	99.7	99.9	99.9	99.9	**99.8**
Test data	Whitespace	80.7	99.8	89.2	87.0	99.9	93.0	91.1
	BreakIter	97.9	90.3	93.9	99.7	98.3	98.9	96.4
	ArkTools	97.5	98.4	97.9	99.3	99.0	99.1	98.5
	LTL-UDE	98.2	99.0	98.6	99.5	98.9	99.2	**98.9**

Table 2: Tokenization results

	(1)	(2)	(3)
Expected	Doz.	im␣Real␣Life	a.␣d.␣gestrigen
Actual	Doz␣.	imRealLife	a.d.gestrigen

Table 3: Tokenization errors

Empiri STTS PoS tags	Freq.	Standard STTS-PoS tags	Freq.
EMOASC	115	PTKANT	42
PTKMA	103	PWAV	39
PTKIFG	99	KOKOM	28
AKW	49	XY	28
HST	46	PDAT	28
ADR	35	VAINF	26
PTKMWL	28	PWS	23
EMOIMG	22	VVIMP	18
URL	18	TRUNC	12
VVPPER	7	KOUI	10
VAPPER	4	PWAT	8
DM	3	VVIZU	7
VMPPER	1	PIDAT	7
ADVART	1	PTKA	5
KOUSPPER	1	APZR	5
ONO	1	VMINF	3
PPERPPER	1	VAPP	3
EML	0	VMPP	1

Table 4: All 18 newly added PoS tags with their frequency of occurrence in the training data compared to the frequency of the 18 least frequent standard STTS PoS tags

(Beißwenger et al., 2015). Table 4 shows all newly added PoS tags with their frequency compared to the least frequent PoS tags that are annotated with a standard STTS PoS tag. As can be seen, 18 PoS tags from the new and standard STTS tagset occur ten times or less. The provided training data thus contains many rare phenomena that cannot be learned from the annotated data alone.

3.2 Implementation

We train a CRF classifier (Lafferty et al., 2001) using the FlexTag tagger (Zesch and Horsmann, 2016) which is based on the DKProTC (Daxenberger et al., 2014) machine learning framework. Our feature set uses a context window of ± 2 tokens, the five-hundred most-frequent character ngrams over all bi, tri and four-grams and boolean features if a token is capitalized, a number, etc.

General Domain Adaptation As the provided training data will not be sufficient to train a competitive model, we decided to apply a domain adaption strategy that has been proposed as an effective method for improving tagging accuracy on social media texts (Ritter et al., 2011; Rehbein, 2013). We closely follow the process outlined in our previous research, where we examined which domain adaption strategies are most likely to improve results (Horsmann and Zesch, 2015). We train a single model on the training data (*CMC* and *Web* subsets) and add additional 100k tokens of newswire text from the Tiger corpus (Brants et al., 2004). To inform the classifier about spelling variations of social media and German morphology we add the following resources:

- *Brown cluster* We create Brown clusters (Brown et al., 1992) from 70 million tokens of German Twitter messages. Spelling variations of the same word form tend to be placed into the same cluster (Ritter et al., 2011), e.g. the unknown word *i-wann* occurs in the same

cluster as the correctly spelled and known word form *irgendwann*. This enables the classifier to learn that *i-wann* and *irgendwann* are distributional similar which provides a bias to assign *i-wann* the same PoS tag as *irgendwann*. We use 1000 clusters and consider words which occur at least 40 times as suggested by Ritter et al. (2011) we provide the resulting bit string in various length as feature to the classifier i.e. 2, 4, 6, ..., 16 (Owoputi et al., 2013) to inform the classifier about (partial) similarity between words.

- *Morphology lexicon* We extract the word class, number and comparative of a word from a German morphology lexicon[1] to inform the classifier about German morphology.

- *PoS dictionary* We create a PoS dictionary which stores the three most frequent PoS tags of a word. We build the dictionary using the Hamburg Dependency Treebank (Foth et al., 2014) which contains STTS annotated text from the technical German website `www.heise.de`. We choose this corpus for its size of almost five million tokens and its technical nature which let it seem more suited for the social media domain than a business newswire corpus.

EmpiriST-specific Adaptation As we have seen in Table 4, some PoS tags are rather rare in the training data and cannot be learned from the data. In order to tackle at least some of those cases, we utilize a post-processing step based on heuristics. For example, all instances of the token *sehr* in the training data are annotated with the same PoS tag. All occurrences of words that start with an @ character are set to *ADR* and those with # are set to *HST*. We also match Urls and Email addresses with regular expressions and assign *URL* or *EML* to them. The word form *sehr* is always assigned *PTKIFG*. Additionally, all words ending in a hyphen are set to *TRUNC*.

We use word lists from Wikipedia and Wiktionary to improve named entity recognition with name lists for person names, cities, countries etc. In those lists, we remove words which occur in the Tiger corpus with a word class other than named entity to filter for words that can occur with other PoS tags, too. Due to unreliable upper- and lowercase usage in social media, we use case-insensitive matching.

A main drawback of adding data from a foreign text domain such as the Tiger corpus is a different annotation scheme and its dominating size that decreases the weight of the EmpiriST training data. This causes a bias for choosing the tags from the bigger Tiger corpus. We attempt to adjust for this bias by adding boolean features if a word can occur with a PoS tag for one of the sparse new word classes to assign a higher weight for choosing a new PoS tag. We added features for instance for focus particles such as *nur, schon, etwas* or words that are verbs merged with personal pronouns such as *schreibste, willste, machste*.

Baseline Systems We use the German model of TreeTagger (Schmid, 1995) as reference point for the performance of our PoS tagger. We report results of applying TreeTagger alone and additionally with our shared-task fitted post-processing to ensure a fair comparison.

3.3 Results & Discussion

Table 5 shows our results on the released gold test data. Each row shows a setting that is applied on the two subsets CMC and Web. For each data set we provide two accuracy values by applying the current setting in its *generic* form and with our shared task-specific (*ST-specific*) post-processing.

The first row in Table 5 shows the performance of the TreeTagger baseline which performs a lot better on the Web data than on CMC data which indicates that Web is much closer to standard German text on which the TreeTagger is known to perform well (Horsmann et al., 2015). The second row shows the performance of tagging the data with a model trained only on the provided EmpiriST training data which performs poorly due to data sparsity. In the third row, we add the foreign domain Tiger corpus which improves accuracy substantially and let our model even beat the baseline on CMC. The subsequent rows show the improvement of adding each of the three resources if added to the EmpiriST and Tiger training data. The morphological lexicon shows the smallest improvements on both data sets. Adding the Brown cluster increases accuracy by 4.6 percent points on the CMC data set but only by 2.5 points on the Web data. We assume that the higher similarity of the Web data to standard German also reduces

[1] `http://www.danielnaber.de/morphologie/`

	CMC		Web		∅	
	Generic	ST-specific	Generic	ST-specific	Generic	ST-specific
TreeTagger	73.8	77.3	91.6	91.8	84.2	84.6
EmpiriST	72.2	73.4	75.5	76.3	73.9	74.9
+Tiger	79.6	80.6	88.8	88.9	84.2	84.8
+Tiger+Brown	84.4	85.2	90.8	90.6	87.6	87.9
+Tiger+MorphLex	81.1	81.5	90.6	90.8	85.9	86.2
+Tiger+PosDict	82.4	83.8	91.0	91.4	86.7	87.6
All resources	85.6	86.1	92.0	92.1	88.8	89.1

Table 5: Results of applying our trained PoS tagger against the released gold test data, we present additional to the overall result the accuracy gain of adding 100k token Tiger and the gains of adding each individual resource compared to training on Empiri+Tiger. We compare our performance against the German TreeTagger model.

PoS tag	Occr.	Acc (%)
PTKMA	85	32.9
FM	49	26.5
VAPPER	4	25.0
VVIMP	32	15.6
PTKIFG	133	15.0
PTKMWL	24	8.3
XY	17	5.9
ADVART	3	0
APPO	1	0
DM	6	0
KOUSPPER	2	0
ONO	2	0
PIDAT	4	0
PPERPPER	1	0

Table 6: Accuracy per word class with an accuracy of less than 50%. PoS tags newly added in the extended STTS tagset are highlighted in grey.

the number of spelling variations in the text which explains the smaller effect of the Brown cluster on the Web data set. The PoS dictionary is with an improvement of 2.5 percent points most effective on the Web data set . If we combine all resources, we improve accuracy on CMC by 8.8 percent points compared to our baseline. On the Web data, the baseline is already quite high, but we still slightly improve by 0.3 points.

To better understand the challenge arising from data sparsity, we show the PoS tags of the test data set which have an accuracy below 50% and are thus especially difficult to tag in Table 6. Noteworthy is that seven word classes have an accuracy of zero. Five of those classes are newly added tags which confirms our assumption that they are too infrequent to be reliably learned.

Figure 1 shows the learning curve of our classifier using both, the provided training data and gold test data. We computed the learning curve as an averaged value with 10fold cross validation. The blue learning curve (triangle) shows the accuracy gain without using any resources. The red curve (square) shows the accuracy gain by additionally adding all of our resources including our shared-task post-processing. The curve without any resources confirms the data sparsity issue. The curve with our resources shows how well our resources compensate data sparsity, but still indicates that more actual training data of the target domain will bring further improvements. Thus, we consider annotating more training data as a promising method to achieve further accuracy improvements.

4 Summary

We presented our approach in the EmpiriST shared task 2015 for the tokenization and PoS tagging of German social media text. We tackled the tokenization task with regular expressions and word lists.

An analysis of the provided training and test data for PoS tagging showes that many of the fine word class distinctions do not occur frequently enough to be learned effectively. We thus utilize foreign domain data, PoS and morphological dictionaries, and clusters of distributional word similarity to overcome sparsity of training data. The added resources show a much higher effectiveness on the CMC data set than on the Web data set,

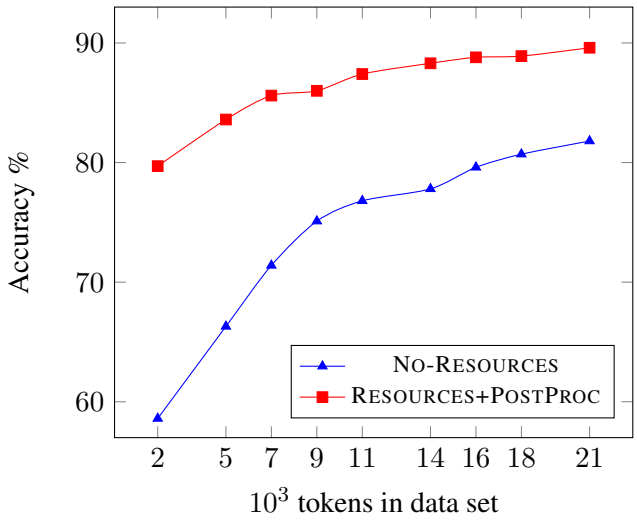

Figure 1: Learning Curve on Empiri-Train and Empiri-Test data averaged in 10fold cross validation, learning curve is shown for using no resources and for using all resources including our post processing.

probably as the Web data set is much closer to standard German text than the CMC data. Furthermore, we presented a learning curve experiment that shows that using more annotated data is likely to yield further improvements.

We make the source code of our experiments publicly available.[2]

References

Michael Beißwenger, Thomas Bartz, Angelika Storrer, and Swantje Westpfahl. 2015. Tagset und Richtlinie für das Part-of-Speech-Tagging von Sprachdaten aus Genres internetbasierter Kommunikation.

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, Social Media and Web Corpora. In *Proceedings of the 10th Web as Corpus Workshop (WAC-X)*, Berlin, Germany.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram corpus version 1.1. *Linguistic Data Consortium*.

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

Peter F Brown, Peter V DeSouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, 18:467–479.

Johannes Daxenberger, Oliver Ferschke, Iryna Gurevych, and Torsten Zesch. 2014. DKPro TC: A java-based framework for supervised learning experiments on textual data. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 61–66, Baltimore, Maryland.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT) at COLING 2014*, pages 1–11, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia.

Kilian A. Foth, Arne Köhn, Niels Beuck, and Wolfgang Menzel. 2014. Because Size Does Matter: The Hamburg Dependency Treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human*

[2] https://github.com/Horsmann/EmpiriSharedTask2015.git

Language Technologies: Short Papers - Volume 2, HLT '11, pages 42–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tobias Horsmann and Torsten Zesch. 2015. Effectiveness of Domain Adaptation Approaches for Social Media PoS Tagging. In *Proceeding of the Second Italian Conference on Computational Linguistics*, pages 166–170, Trento, Italy. Accademia University Press.

Tobias Horsmann, Nicolai Erbs, and Torsten Zesch. 2015. Fast or Accurate ? – A Comparative Evaluation of PoS Tagging Models. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL-2015)*, Essen, Germany.

John D Lafferty, Andrew McCallum, and Fernando C N Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA.

Olutobi Owoputi, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Ines Rehbein. 2013. Fine-Grained POS Tagging of German Tweets. In Iryna Gurevych, Chris Biemann, and Torsten Zesch, editors, *Language Processing and Knowledge in the Web*, volume 8105 of *Lecture Notes in Computer Science*, pages 162–175.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA.

Helmut Schmid. 1995. Improvements In Part-of-Speech Tagging With an Application To German. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. 2004. The Tüba-D/Z Treebank: Annotating German with a Context-Free Backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004)*, Lisbon, Portugal, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L04-1096.

Torsten Zesch and Tobias Horsmann. 2016. Flextag: A highly flexible pos tagging framework. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4259–4263, Portorož, Slovenia. European Language Resources Association (ELRA).