SoMeWeTa: A Part-of-Speech Tagger for German Social Media and Web Texts

Thomas Proisl
Friedrich-Alexander-Universität Erlangen-Nürnberg
Professur für Korpuslinguistik
Bismarckstr. 6, 91054 Erlangen, Germany
thomas.proisl@fau.de

Abstract

Off-the-shelf part-of-speech taggers typically perform relatively poorly on web and social media texts since those domains are quite different from the newspaper articles on which most tagger models are trained. In this paper, we describe SoMeWeTa, a part-of-speech tagger based on the averaged structured perceptron that is capable of domain adaptation and that can use various external resources. We train the tagger on the German web and social media data of the EmpiriST 2015 shared task. Using the TIGER corpus as background data and adding external information about word classes and Brown clusters, we substantially improve on the state of the art for both the web and the social media data sets. The tagger is available as free software.

Keywords: part-of-speech tagging, domain adaptation, evaluation

1. Introduction

Part-of-speech tagging is a core task in natural language processing (NLP) that is important for many subsequent processing steps, e. g. lemmatization, parsing, named-entity recognition or machine translation.

There is already a large number of part-of-speech taggers available that provide pre-trained models for many different languages, including German. Those models are almost always trained on corpora of edited texts written by professional writers, usually newspaper articles. This is also true for German models that are usually trained on the TIGER corpus (Brants et al., 2004). Unfortunately, models trained on newspaper articles perform relatively poorly on web and social media data (Giesbrecht and Evert, 2009).

This is largely due to the many unconventional spelling variants that occur in web and social media texts and that result in a high proportion of out-of-vocabulary (OOV) words on which most taggers perform much worse than on in-vocabulary words. In addition, there are several phenomena in web and social media texts that usually do not occur in edited texts and that cannot be captured properly by most part-of-speech tagsets, including STTS (Schiller et al., 1999). Those phenomena include emoticons, interaction words (e. g. **lach**), hash tags, addressing terms, URLs, onomatopoeia, and colloquial contractions (e. g. *machste*).

One of the aims of the EmpiriST 2015 shared task (Beißwenger et al., 2016)[1] was to improve tokenization and part-of-speech tagging of German computer-mediated communication (CMC) and web corpora. To this end, a gold standard of more than 22,000 tokens (the precise numbers are summarized in Table 1) was annotated with an extended version of STTS called STTS_IBK (Beißwenger et al., 2015). STTS_IBK introduces 18 additional tags, mainly for phenomena typically found in CMC data.

In the present paper, we describe SoMeWeTa (short for Social Media and Web Tagger),[2] a freely available part-of-speech tagger that achieves the best results on the EmpiriST 2015 data published so far.

	CMC	Web
Training	5,109	4,944
Test	5,234	7,568
Total	10,343	12,512

Table 1: Sizes of the EmpiriST training and test sets in tokens.

2. Related Work

Four teams participated in the EmpiriST 2015 shared task. Prange et al. (2016), who won the shared task, use the HMM-based HunPos tagger (Halácsy et al., 2007) and focus on improving the accuracy on out-of-vocabulary words. Their best-performing system is trained on a combination of the TIGER corpus, a version of the TIGER corpus automatically converted to new orthography,[3] the same-domain EmpiriST data set boosted by adding it five times and 34,000 tokens of additional in-domain training data (forum texts, chat and twitter data). Additionally, they use distributional methods to induce a POS lexicon for OOV words which is provided to the tagger at runtime.

Horsmann and Zesch (2016) use the FlexTag tagger (Zesch and Horsmann, 2016) with a conditional random fields (CRF) classifier. Their best-performing system is trained on a combination of the EmpiriST data and 100,000 tokens from the TIGER corpus and uses Brown clusters (Brown et al., 1992), morphological features extracted from Morphy (Lezius, 2000), a POS lexicon extracted from a treebank and lists of named entities as additional resources. The system performs heuristic post-processing steps for hash tags, mentions, URLs, email addresses, the word *sehr* and words ending in hyphens.

[1] https://sites.google.com/site/empirist2015/

[2] https://github.com/tsproisl/SoMeWeTa

[3] This makes Prange et al. (2016) the only team that directly addresses the effects of the German spelling reform.

Remus et al. (2016) build on the GermaNER named entity recognizer (Benikova et al., 2015) to create their CRF sequence tagger. For training their system, they convert the TIGER corpus to the new STTS_IBK tagset and combine it with the same-domain EmpiriST data set. They make use of lists of named entities, similar words from a distributional thesaurus and LDA topic clusters. The system performs some post-processing steps, e. g. for emoticons.

Stemle (2016) builds a tagger based on a long short-term memory (LSTM) recurrent neural network (RNN). The tagger uses word2vec word embeddings and character n-grams of word beginnings and endings and is trained on a combination of the TIGER corpus and the EmpiriST data.

In Table 3, we compare the results of those four teams with our own system which we describe in the following section.

3. System Description

3.1. Learning Algorithm

SoMeWeTa is based on the averaged structured perceptron and uses beam search and an early update strategy. The perceptron, a supervised linear classifier, was introduced by Rosenblatt (1958). It is typically trained by iterating over the training data several times and adjusting the weight vector whenever a misclassification occurs. Freund and Schapire (1999) find that there are two variants that outperform the final weight vector in classification: Implementing a weighted majority voting system based on all states of the weight vector or using the averaged weight vector. We implement averaging. Collins (2002) introduces the structured perceptron that combines the perceptron with a decoding algorithm such as the Viterbi algorithm or beam search to predict, for example, tag sequences for entire sentences. Collins and Roark (2004) suggest the early update strategy for training the structured perceptron, i. e. to stop processing a sentence and to update the weight vector early once it is impossible for the gold sequence of tags to be in the final set of analyses, e. g. because the search beam is too narrow.

3.2. Domain Adaptation

The EmpiriST training set with its 10,000 tokens is too small for training a competitive part-of-speech tagger (cf. Table 3). Therefore, we need to combine it with additional training data, i. e. the TIGER corpus. The TIGER corpus with its 900,000 tokens is much larger. However, it contains texts from a different domain (newspaper articles) and is annotated with its own variant of STTS instead of STTS_IBK. This means that if we simply merge the two data sets, then TIGER will dominate the resulting training corpus and it is rather unlikely that the tagger learns much about web and social media texts and about the novel tags.

A common strategy for dealing with that problem, that is implemented for example by Prange et al. (2016), is boosting, i. e. giving the EmpiriST training set more weight by adding it several times to the training corpus.

We implement a different strategy for domain adaptation. Chelba and Acero (2004) suggest to train a model on the background data, i. e. TIGER, and to use that model as a prior on the weights of the model trained on the in-domain data, i. e. EmpiriST. As Daumé III (2007) points out, that can be achieved for the perceptron by adding the prior weights to the weight vector when making predictions.

{W, N1, N2}.word	W.logfreq
W.prefix	W.lex
{P1, W, N1}.suffix	P2.word + P2.pos
W.shape	P1.word + P1.pos
W.loglength	{P2, P1}.pos
{P2, P1, W, N1, N2}.flags	P2.pos + P1.pos
{P2, P1, W, N1, N2}.brown	P1.pos + W.word

Table 2: Feature templates. W refers to the current token, P2 and P1 to the previous two tokens, N1 and N2 to the next two tokens.

3.3. Feature Templates

We forgo meticulous feature engineering and use a fairly standard set of features that is summarized in Table 2. The word, prefix and suffix features use the lower cased version of the token. For the shape feature, we map all characters to a character type: Upper case letters to "X", lower case letters to "x" and digits to "d"; all other characters remain the same. We limit the number of consecutive characters of the same type to 4 (this means that "Computer" becomes "Xxxxx"). The loglength feature is the logarithm of the token's length in characters, rounded to an integer. The flag features indicate:

- If all characters of the token are alphabetic, numeric or punctuation;
- if the token is in lower, upper or title case;
- if the word is an email address, an XML tag, a URL, a mention (*@peter*), a hashtag, an interaction word (**grins**), an emoticon, an ordinal number or a number.

The brown feature contains the Brown cluster of the word; the logfreq feature is the logarithm of the frequency of the word according to the Brown cluster file, rounded to an integer. Lex is used for features found in the additional lexicon – in our case major word classes from Morphy and/or capitalization information.

3.4. External Resources

In addition to the EmpiriST training data, we use the following external resources:

- The TIGER corpus is used as background data in the domain adaptation process described above.
- We use DECOW14 (Schäfer, 2015; Schäfer and Bildhauer, 2012) to extract capitalization features, i. e. flags that indicate how a word is capitalized in the majority of its occurrences, and 1,000 Brown clusters (Brown et al., 1992).[4]
- We use information about major word classes from Morphy (Lezius, 2000).[5]

[4]We use the implementation by Liang (2005): `https://github.com/percyliang/brown-cluster/`.

[5]Extracted following these instructions: `http://www.danielnaber.de/morphologie/`.

4. Results and Error Analysis

4.1. Data Preprocessing

The part-of-speech annotation in TIGER differs in some details from the original STTS. It does not distinguish between indefinite pronouns in attributive function with and without determiner (STTS: PIDAT vs. PIAT; TIGER: PIAT), tags prepositions as ADV when they modify numerals and uses the tag PROAV for pronominal adverbs instead of PAV (Smith, 2003, 13). STTS_IBK, however, is an extension of the original STTS and not of the TIGER variant. Therefore, we automatically replace all instances of the tag PROAV with the original STTS tag PAV. We do not address TIGER's other two deviations from STTS.

While TIGER is annotated with sentence boundaries, the EmpiriST data set is not. Therefore, we automatically introduce sentence boundaries using the sentence splitter that is part of the SoMaJo tokenizer for German web and social media texts (Proisl and Uhrig, 2016).[6]

In the EmpiriST data set, emojis have been replaced with textual representations, e. g. emojiQloudlyCryingFace. Since we want our tagger model to be applicable to real-world data, we reintroduce the actual Unicode characters.

4.2. Results

We run evaluation experiments for different combinations of external resources. In all evaluation settings, we train the tagger for ten iterations. Since SoMeWeTa shuffles the training data after each iteration, the training process is subject to some amount of random variation. To account for that and to give a realistic representation of the tagger's performance, we run all evaluation experiments ten times and report the mean of the ten runs $\pm$ two standard deviations. We follow Beißwenger et al. (2016) and report the tagging accuracies on the two EmpiriST data sets (CMC and web data) as well as their macro-average for each evaluation setting (Table 3). We also include figures for in-vocabulary and out-of-vocabulary accuracies.

As points of reference, we include results for scenarios where the tagger is trained only on TIGER or only on the EmpiriST data. In all other evaluation settings, TIGER is used as background data in the domain adaptation process described above.

As we can see, simply using the domain adaptation strategy described above without any additional external resources already yields competitive results (89.11% overall acc.). Both the word class information from Morphy and the Brown clusters extracted from DECOW14 prove to be useful individually and their combination boosts the accuracy even further (91.06% overall), substantially improving on the previous state of the art (Prange et al., 2016) on both subsets. The impact of the capitalization features extracted from DECOW14, on the other hand, seems to be rather limited or even counterproductive in some cases.

As an alternative way for arriving at a single accuracy figure for the data, we combine the CMC and Web test sets into a single test set. On this unified test set, the system by Prange et al. (2016) achieves 91.01% accuracy. SoMeWeTa with the TEB+Mor setting achieves a mean accuracy of 91.55% $\pm$0.18 – a statistically significant improvement (McNemar's test, $p < 0.05$).

We also run experiments where we only use the same-domain EmpiriST data for training. Interestingly, this only leads to better results for the CMC data but not for the web data. It seems that for the web data, which is closer in nature to the newspaper articles in TIGER than the CMC data, the advantage of having a larger corpus for adapting to the novelties of the tagset outweighs the heterogeneity of the data. For the CMC data, on the other hand, being able to adjust to the very different domain by means of a more homogeneous corpus seems to be more important than having more data on the new tags.

4.3. Error Analysis

For the error analysis, we use the run of the TEB+Mor model that is closest to the means reported in Table 3.

Surprisingly, the largest source of errors in the CMC part is the confusion of the two tags $(and $. (cf. Table 4). The reason for this is a peculiarity in the gold data with regard to the colon (:). In the TIGER corpus and in the web part of the EmpiriST data, almost all colons are annotated as $. (97.1% in TIGER and 100% in the web part). In the CMC part, however, only 31.2% of all colons are annotated as $., whereas 68.8% are annotated as $(.

Another interesting source of errors is the frequent misclassification of graphical emoticons (EMOIMG). As it turns out, these errors are mainly due to the fact that there are several sequences of multiple emoticons in the test data, but not in the training data where emoticons always occur in isolation. Other major sources of errors in the CMC and web parts (Table 5) are the confusion of NN and NE, misclassifications related to the new particle tags PTKIFG, PTKMA and PTKMWL and misclassifications within the verb tags.

A visual overview of frequent classes of errors is given in Figure 1. For this visualization, we map the tags to the coarser tagset used in the Universal Dependencies project (Nivre et al., 2016). Note that the new particle tags PTKIFG, PTKMA and PTKMWL get mapped to ADV. We can clearly see that the four major sources of errors are misclassifications within the ADV and VERB tags and the confusion of NOUN and PROPN.[7]

4.4. Post-analysis experiments

As noted in the previous section, there are no sequences of graphical emoticons in the training data. As a consequence, sequences of emoticons are frequently misclassified. Other problematic phenomena we found when using the tagger to annotate a corpus of tweets were sequences of hashtags and mentions that were embedded in the syntactic structure of the sentence.

As a countermeasure to these sources of errors, we created a small file with additional training data taken from a collection of tweets about the German federal election.

[6] https://github.com/tsproisl/SoMaJo

[7] When we map the gold standard and the tagger output to this reduced tagset we get rid of within-class misclassifications and trade tagset granularity for higher accuracy. With its output mapped to UD POS, SoMeWeTa achieves the following accuracies with the TEB+Mor model: 92.69% $\pm$0.19 on CMC, 95.81% $\pm$0.20 on Web, 94.25% overall and 94.54% $\pm$0.12 on the unified test set.

System	CMC all	CMC IV	CMC OOV	Web all	Web IV	Web OOV	Overall
Prange et al. (2016)	87.33	–	–	93.55	–	–	90.44
Horsmann and Zesch (2016)	86.07	–	–	92.10	–	–	89.09
Remus et al. (2016)	84.22	–	–	93.27	–	–	88.75
Stemle (2016)	85.42	–	–	90.63	–	–	88.03
only TIGER	72.24 ±0.31	83.96 ±0.36	33.42 ±0.52	92.03 ±0.21	94.58 ±0.19	77.48 ±1.11	82.14
only EmpiriST	79.72 ±0.40	86.80 ±0.53	62.09 ±1.23	83.58 ±0.64	89.71 ±0.63	73.49 ±0.95	81.65
TIGER+EmpiriST (TE)	85.78 ±0.40	89.07 ±0.45	65.44 ±1.00	92.43 ±0.27	94.40 ±0.15	79.46 ±1.21	89.11
TE+Mor	87.50 ±0.39	90.17 ±0.42	70.96 ±0.88	93.17 ±0.28	94.84 ±0.21	82.19 ±1.27	90.34
TE+Brown (TEB)	88.16 ±0.41*	90.26 ±0.31	75.15 ±1.49	93.72 ±0.17	95.26 ±0.21	**83.58** ±0.91	90.94
TEB+cap	88.20 ±0.29	90.22 ±0.26	75.73 ±1.31	93.58 ±0.18	95.11 ±0.11	83.53 ±0.85	90.89
TEB+Mor	88.37 ±0.26*	90.36 ±0.22	76.06 ±1.27	**93.75** ±0.24	**95.31** ±0.24	83.54 ±0.95	**91.06**
TEB+Mor+cap	88.32 ±0.42*	90.41 ±0.34	75.36 ±1.33	93.73 ±0.18	95.29 ±0.20	83.50 ±0.69	91.03
$TE_{CMC}B$+Mor	88.61 ±0.45**	90.67 ±0.36	**76.08** ±1.53	92.97 ±0.27	94.59 ±0.29	82.97 ±0.91	90.79
$TE_{Web}B$+Mor	76.26 ±0.58	85.47 ±0.74	43.95 ±1.02	93.68 ±0.30	95.29 ±0.25	83.36 ±1.05	84.97
$TE_{CMC}B$+Mor+cap	**88.69** ±0.44**	**90.88** ±0.40	75.33 ±1.61	93.04 ±0.18	94.60 ±0.17	83.36 ±0.44	90.87
$TE_{Web}B$+Mor+cap	76.36 ±0.32	85.46 ±0.27	44.44 ±1.61	93.57 ±0.26	95.27 ±0.25	82.68 ±0.90	84.97

Table 3: Evaluation results. We report the average accuracy of ten runs ± two standard deviations. Abbreviations: TE = TIGER+EmpiriST, TEB = TIGER+EmpiriST+Brown, Mor = word class information from Morphy, cap = capitalization features from DECOW14. Stars indicate a statistically significant improvement on Prange et al. (2016) (McNemar's test, $p < 0.05, 0.01, 0.001$).

tag	freq	err	most frequent confusions
$(	343	53	$. (42), KON (7), XY (3)
NN	696	47	NE (12), FM (7), ADJA (6)
ADV	268	46	PTKMA (14), PIS (6), PTKIFG (6)
PTKIFG	72	46	ADV (32), ADJD (8), PIS (3)
NE	230	40	NN (20), ADJA (4), ADJD (3), ITJ (2)
EMOIMG	63	34	$((21), XY (6), $. (5),
ADJD	187	32	ADV (9), VVPP (7), ADJA (3), NN (3)
ITJ	99	21	NN (5), PTKANT (3), ADJA (2)
PTKMA	74	21	ADV (15), ADJD (3)
VVFIN	183	20	VVINF (8), NN (5), VVIMP (4)
ADJA	149	18	NN (6), NE (4), FM (2), PIAT (2)
VVIMP	20	16	VVFIN (5), NN (4), ART (2), ITJ (2)
XY	14	12	PPER (5), $((2), ITJ (2)
PTKVZ	40	11	APPR (3), ADV (2), PTKIFG (2)
VVINF	87	11	VVFIN (5), NN (3), VVPP (3)
KOKOM	21	10	APPR (5), FM (2), PWAV (2)
KON	112	10	ADV (5), VAFIN (2)

Table 4: Errors in CMC (10 or more errors per gold tag)

tag	freq	err	most frequent confusions
NN	1661	96	NE (74), FM (9), ADJA (3), ADV (3)
NE	252	43	NN (39)
ADV	309	42	PTKIFG (13), PTKMA (12), PIS (4)
VVFIN	250	37	VVINF (17), VVPP (16), NN (2)
PTKIFG	61	36	ADV (28), ADJD (6)
FM	43	24	NE (14), NN (2), VAFIN (2)
ADJA	498	16	NN (4), FM (3), ADJD (2)
PTKMWL	14	14	ADV (9), ADJD (2)
APPR	583	13	KOKOM (6), ADV (4)
ART	729	12	PRELS (7), PDS (5)
PIAT	79	10	PIS (6), ADJA (2)
VVINF	125	10	VVFIN (6), NN (4)
VVPP	125	10	ADJD (7)

Table 5: Errors in Web (10 or more errors per gold tag)

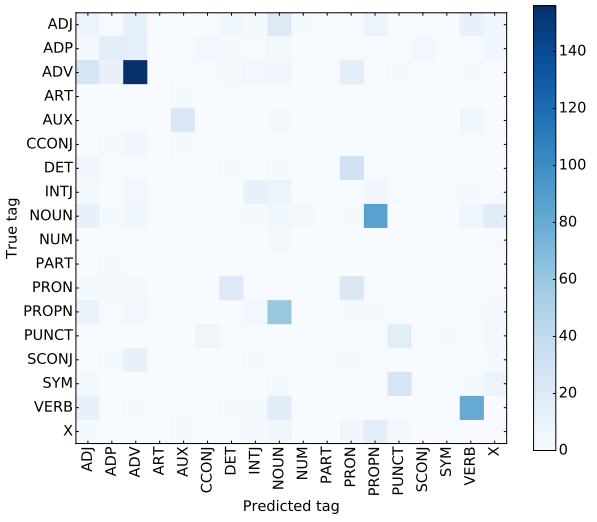

Figure 1: Errors made on the unified test set. For clarity of presentation, we map the tags to UD UPOS tags, i. e. errors on the diagonal indicate misclassifications between tags that get mapped to the same UD UPOS tag.

This file contains 126 tokens in 13 "sentences". The additional training data file covers phenomena typically found in computer-mediated communication. Therefore, its addition leads to improvements for the CMC subset while not affecting the Web subset. The best model achieves an overall accuracy of 91.42% (cf. Table 6).

When we use the TEB+Mor+add model on the unified test set mentioned in Section 4.2., the tagger achieves a mean accuracy of 91.84% ±0.17. This is a statistically significant improvement on Prange et al. (2016) (McNemar's test, $p < 0.001$).

System	CMC all	CMC IV	CMC OOV	Web all	Web IV	Web OOV	Overall
TEB+Mor+add	89.08 ±0.25***	90.95 ±0.27	77.41 ±1.14	**93.75** ±0.26	**95.34** ±0.25	83.31 ±0.63	**91.42**
TEB+Mor+cap+add	88.84 ±0.29***	90.84 ±0.27	76.39 ±1.08	93.71 ±0.29	95.29 ±0.29	83.34 ±0.67	91.28
TE_{CMC}B+Mor+add	**89.17** ±0.38***	91.08 ±0.35	**77.48** ±1.47	93.10 ±0.24	94.62 ±0.19	83.69 ±0.89	91.14
TE_{Web}B+Mor+add	77.82 ±0.59	85.56 ±0.27	50.12 ±2.57	93.67 ±0.19	95.25 ±0.18	83.49 ±0.93	85.75
TE_{CMC}B+Mor+cap+add	89.10 ±0.40***	**91.18** ±0.38	76.36 ±0.94	93.09 ±0.24	94.58 ±0.22	**83.85** ±0.83	91.10
TE_{Web}B+Mor+cap+add	78.16 ±0.35	85.71 ±0.40	51.18 ±2.37	93.62 ±0.32	95.27 ±0.26	82.97 ±1.09	85.89

Table 6: Evaluation results with additional training data (+add). We report the average accuracy of ten runs ± two standard deviations. Abbreviations as in Table 3. Stars indicate a statistically significant improvement on Prange et al. (2016) (McNemar's test, $p < 0.05, 0.01, 0.001$).

5. Conclusion

SoMeWeTa is based on a relatively simple linear classifier in combination with a suitable strategy for domain adaptation. Valuable additional resources are a lexicon with word class information and Brown clusters extracted from a web corpus. Although we do not fine-tune the features or apply post-processing steps that address common errors, SoMeWeTa substantially improves on the current state of the art. Given that other systems, e. g. Horsmann and Zesch (2016), use very similar external resources, the key to SoMeWeTa's superior performance seems to be the more sophisticated domain adaptation process.

Both the tagger and a model trained on the entire EmpiriST data set are freely available.[8]

6. Bibliographical References

Beißwenger, M., Bartz, T., Storrer, A., and Westpfahl, S. (2015). Tagset und Richtlinie für das Part-of-Speech-Tagging von Sprachdaten aus Genres internetbasierter Kommunikation. Guideline document.

Beißwenger, M., Bartsch, S., Evert, S., and Würzner, K.-M. (2016). EmpiriST 2015: A shared task on the automatic linguistic annotation of computer-mediated communication and web corpora. In Paul Cook, et al., editors, *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 44–56, Berlin. Association for Computational Linguistics.

Benikova, D., Yimam, S. M., Santhanam, P., and Biemann, C. (2015). GermaNER: Free open German named entity recognition tool. In Bernhard Fisseni, et al., editors, *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL 2015)*, pages 31–38, Duisburg-Essen. GSCL.

Brants, S., Dipper, S., Eisenberg, P., Hansen-Schirra, S., König, E., Lezius, W., Rohrer, C., Smith, G., and Uszkoreit, H. (2004). TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation*, 2(4):597–620.

Brown, P. F., Pietra, V. J. D., de Souza, P. V., Lai, J. C., and Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Chelba, C. and Acero, A. (2004). Adaptation of maximum entropy capitalizer: Little data can help a lot. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 285–292, Barcelona. Association for Computational Linguistics.

Collins, M. and Roark, B. (2004). Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pages 111–118, Barcelona. Association for Computational Linguistics.

Collins, M. (2002). Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1–8, Philadelphia, PA. Association for Computational Linguistics.

Daumé III, H. (2007). Frustratingly easy domain adaptation. In Annie Zaenen et al., editors, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 256–263, Prague. Association of Computational Linguistics.

Freund, Y. and Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

Giesbrecht, E. and Evert, S. (2009). Is part-of-speech tagging a solved task? An evaluation of pos taggers for the German web as corpus. In Iñaki Alegria, et al., editors, *Proceedings of the Fifth Web as Corpus Workshop (WAC5)*, pages 27–35, San Sebastián. Elhuyar Fundazioa.

Halácsy, P., Kornai, A., and Oravecz, C. (2007). Hunpos – an open source trigram tagger. In John A. Carroll, et al., editors, *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 209–212, Prague. Association for Computational Linguistics.

Horsmann, T. and Zesch, T. (2016). LTL-UDE @ EmpiriST 2015: Tokenization and pos tagging of social media text. In Paul Cook, et al., editors, *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 120–126, Berlin. Association for Computational Linguistics.

Lezius, W. (2000). Morphy – German morphology, part-of-speech tagging and applications. In Ulrich Heid, et al., editors, *Proceedings of the 9th EURALEX International Congress*, pages 619–623, Stuttgart. Institut für Maschinelle Sprachverarbeitung.

[8] https://github.com/tsproisl/SoMeWeTa

Liang, P. (2005). Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal Dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari, et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož. European Language Resources Association.

Prange, J., Horbach, A., and Thater, S. (2016). UdS-(retrain|distributional|surface): Improving pos tagging for oov words in German cmc and web data. In Paul Cook, et al., editors, *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 63–71, Berlin. Association for Computational Linguistics.

Proisl, T. and Uhrig, P. (2016). SoMaJo: State-of-the-art tokenization for German web and social media texts. In Paul Cook, et al., editors, *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 57–62, Berlin. Association for Computational Linguistics.

Remus, S., Hintz, G., Biemann, C., Meyer, C. M., Benikova, D., Eckle-Kohler, J., Mieskes, M., and Arnold, T. (2016). EmpiriST: AIPHES – Robust tokenization and pos-tagging for different genres. In Paul Cook, et al., editors, *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 106–114, Berlin. Association for Computational Linguistics.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Schiller, A., Teufel, S., Stöckert, C., and Thielen, C. (1999). Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset). Technical report, IMS Stuttgart, SfS Tübingen.

Schäfer, R. and Bildhauer, F. (2012). Building large corpora from the web using a new efficient tool chain. In Nicoletta Calzolari, et al., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 486–493, Istanbul. European Language Resources Association.

Schäfer, R. (2015). Processing and querying large web corpora with the COW14 architecture. In Piotr Bański, et al., editors, *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, pages 28–34, Lancaster. UCREL, IDS.

Smith, G. (2003). A brief introduction to the TIGER treebank, version 1. Technical report, Universität Potsdam.

Stemle, E. (2016). bot.zen @ EmpiriST 2015 – A minimally-deep learning pos-tagger (trained for German cmc and web data). In Paul Cook, et al., editors, *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 115–119, Berlin. Association for Computational Linguistics.

Zesch, T. and Horsmann, T. (2016). FlexTag: A highly flexible pos tagging framework. In Nicoletta Calzolari, et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4259–4263, Portorož. European Language Resources Association.

7. Language Resource References

Michael Beißwenger and Sabine Bartsch and Stefan Evert and Kay-Michael Würzner. (2016). *EmpiriST 2015 Gold Standard*. `https://sites.google.com/site/empirist2015/home/gold/`.

Felix Bildhauer and Roland Schäfer. (2014). *DECOW14*. COW – Corpora from the Web, `http://corporafromtheweb.org/decow14/`.

Sabine Brants and Stefanie Dipper and Peter Eisenberg and Silvia Hansen and Esther König and Wolfgang Lezius and Christian Rohrer and George Smith and Hans Uszkoreit. (2004). *TIGER Corpus*. IMS Stuttgart, `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html`.

Percy Liang. (2012). *Brown-cluster*. `https://github.com/percyliang/brown-cluster`.

Thomas Proisl and Peter Uhrig. (2016). *SoMaJo*. `https://github.com/tsproisl/SoMaJo`.

Thomas Proisl. (2017). *SoMeWeTa*. `https://github.com/tsproisl/SoMeWeTa`.