Fast or Accurate? – A Comparative Evaluation of PoS Tagging Models

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

Abstract

We perform a comparison of 27 PoS tagger models for English and German offered by 9 different implementations. By evaluating on a mix of corpora from different domains, we simulate a black-box usage where researchers select a tagger (because of popularity, ease of use, etc.) and apply it to all sorts of text. Surprisingly, a manually created rule-based model outperforms all learned models with respect to accuracy and speed. Within the group of learned models, we find the expected trade-off between fast models with relatively low accuracy and slower models with higher accuracy. Our evaluation provides researchers with a basis for selecting taggers according to their needs.

1 Introduction

Part-of-Speech (PoS) tagging is one of the most important steps in Natural Language Processing (NLP). Consequently, researchers can choose from a wide range of available PoS taggers, popular choices include TreeTagger (Schmid, 1995), Stanford Tagger (Toutanova et al., 2003), or ClearNLP (Choi and Palmer, 2012). The decision for a certain tool is mainly influenced by tagging accuracy, but other practical issues like ease of use, speed, applicability to target language and domain, or availability for a certain hardware platform might also play a role.

In this paper, we focus on tagging accuracy vs. speed and perform a comparative evaluation of 27 tagging models for English and German, offered by 9 different PoS tagger implementations. We evaluate on a range of English and German corpora from three different broad domains (formal writing, speech transcripts, and social media).

To our knowledge, this is the most comprehensive evaluation to date. Giesbrecht and Evert (2009) compared German models of five PoS taggers and Miguel and Roxas (2007) compared four Tagalog taggers on a single corpus.

PoS tagging A PoS tagger is an application that assigns the word class (i.e. the PoS tag) to each token in a sentence. PoS taggers can loosely be categorized into unsupervised, supervised, and rule-based taggers.

Unsupervised taggers (Goldwater and Griffiths, 2007; Biemann, 2006; Das and Petrov, 2011) analyze large quantities of plain text and group words by their context similarity. The assumption is that words that are grouped together share the same word class. However, this word class is not made explicit in this case, which is why unsupervised taggers are rarely used on their own but usually added as features in a supervised setting (Ritter et al., 2011).

Supervised taggers are machine learning applications that require manually annotated training data. The tagger takes the annotated text and extracts text properties (so called *features*) that are provided to the machine learning classifier which learns a model that maps the feature representation of tokens to the corresponding PoS tags. When running the tagger, the same feature representation is extracted from the raw input text and the trained model is applied to select a tag for every token based on the feature values. A model is thus best applied to input text that is as similar to the training data as possible. In case of a mismatch, e.g. a model trained on newswire applied to speech transcripts, the extracted feature values might not match with the expected ones. As a consequence, the tagging accuracy is considerably reduced.

Rule-based taggers utilize sets of patterns or rules to assign tags. In principle, they are very similar to the supervised taggers, only that the underlying model is not automatically learned but hand-curated.

Research question In this paper, we focus on supervised and rule-based taggers, and ask the question: which is the best tagger? However, as we have learned above, supervised taggers are machine learning applications that use a tagging model. Thus, many taggers come with several models that are optimized for different domains or offer trade-offs between accuracy and speed. Thus, the statement *Tagger X performs well* needs to be rephrased as *Tagger X using model Y performs well on corpus Z*.

As the performance of a tagger relies on a complex mix of machine learning, feature representation, and the applied external resources, we cannot analytically decide which tagger is the best. Instead, we perform an empirical evaluation that will provide researchers with a sound basis for their choice of a PoS tagger.

2 Experimental setup

In our experiment, we want to evaluate the tagger models of various PoS tagger implementations against a large number of corpora from various text domains. We base our experiments on the DKPro Core framework (Eckart de Castilho and Gurevych, 2014) that is based on UIMA (Ferrucci and Lally, 2004). DKPro Core provides wrappers for a wide range of taggers shielding the user from the intricate details of installing and invoking the taggers and offering simple, unified usage by providing a shared interface. A UIMA workflow follows a pipeline principle where documents are passed through and processed by an arbitrary number of processing components.

2.1 Processing pipeline

In our setup, each corpus is read and transformed into the internal representation of DKPro Core which is based on stand-off annotations. The tagging is done by a wrapper-component that encapsulates the PoS taggers and allows for using all taggers over a common interface. The wrapper transforms the internal representation of the text into the format which the tagger requires and transforms the tagged text back into the internal representation for further processing. We then apply a post-processing step (Ritter et al., 2011) that uses regular expressions to recognize and correctly tag special entities like email addresses, URLs, and Twitter-specific phenomena like hashtags, at-mentions, and retweets. A final evaluation component compares the assigned tags to the gold tags from the corpus.

Directly before and after the tagger component, we inject time measuring components in order to ensure that only the actual time spent for tagging is measured. However, our measuring includes the time that the wrapper needs to feed the data to the underlying tagger implementation. In case of Java taggers, this is usually just a method call, but in case of wrapped C binaries there might be a considerable overhead. Thus, the runtime reported in this study might differ than when running a tagger without the wrapper.

A further issue that might affect the time measurement is document size. Some taggers are fastest when fed with small chunks of data, while others are optimized for processing large documents as a whole. In order to account for this difference, we run all experiments twice: (i) with each sentence as a unit of processing, and (ii) the entire corpus as a unit of processing. We then report the run that takes less time.[1]

2.2 Tagger implementations and models

We now describe the PoS taggers and their models used in this study (see Table 1 for an overview). If available, we provide information about the domain of the training data that were used to train the models.

Arktools (Owoputi et al., 2013) is tailored to tag social media messages. Three models are available, the first one is trained on Twitter data by (Gimpel et al., 2011; Owoputi et al., 2013), which use the coarse-grained Gimpel tagset. The other two use the Penn Treebank (PTB) tagset and are trained on annotated IRC chat data by (Forsyth and Martell, 2007) and Tweets by (Ritter et al., 2011).

ClearNLP (Choi and Palmer, 2012) provides two English models. One trained on medical text and one trained on a mixture of text from various genres that is mostly news-related.

Hepple (Hepple, 2000) is a rule-based tagger similar to the Brill-Tagger (Brill, 1992).

HunPos (Halácsy et al., 2007) is an open-source reimplementation of the TNT tagger (Brants, 2000). Newswire models are available for English trained on the WSJ and for German trained on the Tiger corpus.

LBJ (Roth and Zelenko, 1998) provides a model for English trained on newswire text.

[1] Note that the accuracy in both cases is always equal, as the same sentences are tagged.

Tool	Language	Trained on	Modelname	Tagset	Domain	Abbr.
Ark	en	Owoputi	default	Gimpel	social	A-1
		Irc	irc	PTB-NPS	social	A-2
		Ritter	ritter	PTB-RIT	social	A-3
ClearNLP	en	Medical text	mayo	PTB	clinical	C-1
		OntoNotes	ontonotes	PTB	news	C-2
Hepple	en	*rule-based*		PTB	-	Hepple
HunPos	en	WSJ	wsj	PTB	news	Hun
	de	Tiger	tiger	STTS	news	
Mate	en	CoNLL2009	conll2009	PTB	mixed	Mate
	de	Tiger	tiger	STTS	news	
Lbj	en	WSJ	-	PTB	news	Lbj
OpenNLP	en	*unknown*	maxent	PTB	*unknown*	O-1
		unknown	perceptron	PTB	*unknown*	O-2
	de	Tiger	maxent	STTS	news	O-3
		Tiger	perceptron	STTS	news	O-4
Stanford	en	WSJ	bidirectional-distsim	PTB	news	St-1
		WSJ	caseless-left3w.-distsim	PTB	news	St-2
		unknown	fast	PTB	*unknown*	St-3
		Twitter/WSJ	twitter-fast	PTB-RIT	mixed	St-4
		Twitter/WSJ	twitter	PTB-RIT	mixed	St-5
		WSJ	wsj-0-18-caseless-left3w.-distsim	PTB	news	St-6
	de	Negra	dewac	STTS	news	St-7
		unknown	fast-caseless	STTS	news	St-8
		Negra	fast	STTS	news	St-9
		Negra	hgc	STTS	news	St-10
TreeTagger	en	*unknown*	le	PTB-TT	news	Tree
	de	*unknown*	le	STTS	news	

Table 1: Tagger models used in our experiments.

Mate (Björkelund et al., 2010) provides an English model trained on CoNLL2009 (Hajič et al., 2009) and a German model trained on the Tiger newswire corpus.

OpenNLP is an Apache project that provides a wide range of NLP tools including a tagger.[2] It provides models for English and German based on two different classifiers (Maximum Entropy and Perceptron). The German models are trained on the Tiger corpus. We could not find any information about the training data of the English models.

Stanford (Toutanova et al., 2003) provides several English and German models for their tagger. The models differ with respect to lowercasing of all tokens, adding distributional knowledge, or using a bidirectional model. Two social media models are trained by Derczynski et al. (2013).[3] The origin of some models is unknown.

TreeTagger (Schmid, 1994; Schmid, 1995) provides an English model trained on the Penn-Treebank and further proprietary resources as well as a German model for which little information is available.

[2] https://opennlp.apache.org

[3] https://gate.ac.uk/wiki/twitter-postagger.html

2.3 Tagsets

A tagset is a collection of labels which represent word classes. A coarse-grained tagset might only distinguish main word classes such as adjectives or verbs, while more fine-grained tagsets also make distinctions within the broad word classes, e.g. distinguishing between verbs in present and past tense.

Many English models are trained on corpora annotated with the PTB tagset, which distinguishes 48 tags (Marcus et al., 1993). Some models add additional tags to the PTB in order to distinguish further language phenomena. Schmid (1994) assigns the inflection forms of the words *be, do, have* an own tag instead of the default verb tags. Likewise, the word *that* is tagged with an own tag if it occurs as preposition. Ritter et al. (2011) added four additional tags to label the phenomenons that frequently occur in Twitter messages like hashtags or URLs. Forsyth and Martell (2007) prefix PTB tags with an extra character in case the word-form

	Domain	Corpus	Tokens in (10^3)	Tagset
en	written	BNC-News	100	C5
		Brown	1,100	Brown
		MASC-Essay	37	PTB
		MASC-Fiction	38	PTB
		MASC-Govern.	28	PTB
		MASC-Journal	24	PTB
		MASC-Non-Fict.	30	PTB
		MASC-TechDoc	23	PTB
		MASC-Travel	28	PTB
	spoken	MASC-Convers.	100	PTB
		MASC-Court	35	PTB
		MASC-Debate	36	PTB
		MASC-F2Face	28	PTB
		MASC-Teleph.	5	PTB
		Switchboard	2,100	PTB
	social	Gimpel	27	Gimpel
		MASC-Blog	33	PTB
		MASC-Email	63	PTB
		MASC-Twitter	29	PTB
de	written	Tüba-DZ	1,500	STTS
	social	Twitter-Reh	20	STTS

Table 2: Corpora used in our experiments.

is misspelled.

Other tagsets used in the evaluation corpora are Brown (Nelson Francis and Kuçera, 1964) and C5 (BNC) as well as the coarse-grained Gimpel tagset with 25 tags specialized on social media. In German, the *Stuttgart-Tübingen-TagSet* (STTS) with 54 tags is exclusively used.

If a model trained on a corpus with a certain tagset is evaluated on a corpus using a second tagset, this mismatch will result in artificially low accuracy. Thus, we map the fine-grained tags to the coarse grained *universal tagset* (Petrov et al., 2012) as implemented by DKPro Core. Obviously, subtle distinctions between similar tags will be lost in the process, but for many downstream applications fine-grained distinctions between sub-tags of the same word class are not important anyway. Thus, the coarse-grained accuracy gives a good approximation of the expected tagging quality.

2.4 Corpora

Table 2 gives an overview of the corpora used in our evaluation. We partition the English corpora into three broad domains: (i) formal writing, (ii) speech transcripts, and (iii) social media. We choose this partitioning to challenge the taggers with inherent different contents. For German, we could only find corpora for the written and social media domains.

English The first set of corpora contains formal writing, e.g. news articles, fiction, or technical documentation. We use subset of the newswire text from the British National Corpus[4], the Brown corpus (Nelson Francis and Kuçera, 1964) which contains American English of the 1960's, and eight subsections of the MASC (Ide et al., 2010) corpus with text from several written subdomains. The second set contains transcripts of spoken language. We use the Switchboard (Marcus et al., 1993) corpus (telephone conversations) and five speech-related subsections of MASC. The third set contains social media messages that combine properties of written and spoken language. Social media is characterized by its high vocabulary heterogeneity and many domain-specific tokens as emoticons, URLs, or email addresses which are likely to be out-of-vocabulary for most tagger models. We use four subsections of MASC as well as annotated Twitter messages by Gimpel et al. (2011).

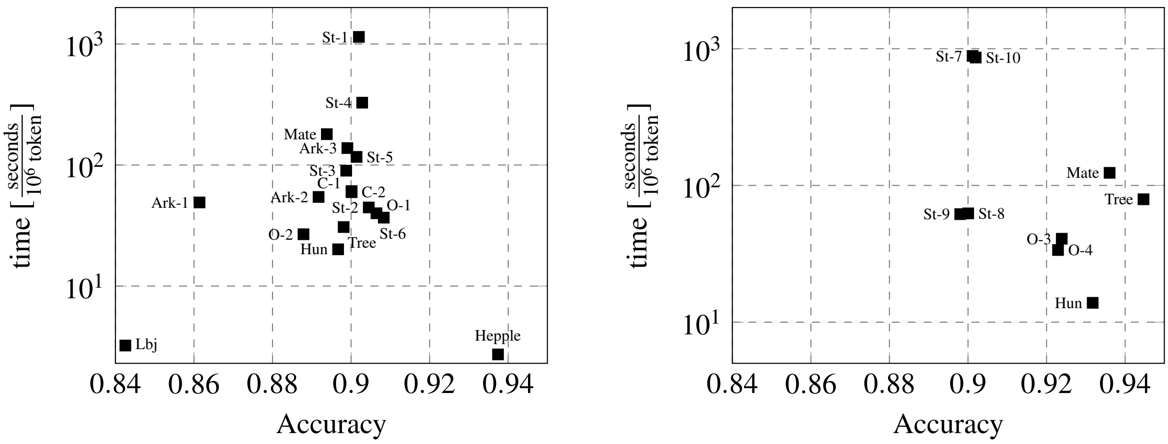
In order to avoid testing on the training data, we exclude other available PoS-annotated corpora like the WSJ corpus (Marcus et al., 1993), the Twitter corpus by Ritter et al. (2011), or the IRC chat corpus (Forsyth and Martell, 2007), as many of the models have been trained using those corpora. As the provenance of some models is unknown, their results should be treated with caution as we might still be testing on the training data here.

German We use the STTS-annotated Tüba-DZ corpus (Telljohann et al., 2004) based on the German newspaper *die tageszeitung* and the Twitter-Reh corpus (Rehbein, 2013) of German Tweets annotated with an Twitter-specific extension of STTS following Ritter et al. (2011). We exclude the Tiger corpus (Brants et al., 2004) and the Negra corpus (Skut et al., 1998) as all German models are trained on one of the two.

3 Results and Analysis

After evaluating all tagger models on all corpora we obtain the results shown for English in Figure 1a and for German in Figure 1b. The x-axis shows the macro-averaged tagging accuracy based on the coarse-grained universal tagset. As discussed above, we cannot use fine-grained tags for evaluation, because of frequent mismatches between the tagset used by the tagger and the tagset used in the evaluation corpus. The y-axis shows the normalized processing time in seconds per million tokens.

[4] http://www.natcorp.ox.ac.uk/



(a) English

(b) German

Figure 1: Macro-averaged results over all corpora.

Of course the hardware[5] will influence the absolute time spent on the task, but the relative differences between the models are of greater importance here.

In general, we observe the expected trade-off between (i) high-accuracy taggers that invest a lot of processing into feature extraction or more sophisticated classifiers and are thus slower, and (ii) high-speed taggers that can process much more tokens in the same time at the cost of accuracy. For example, on the English corpora *Lbj* is extremely fast, but reaches only a low accuracy while *St-6* or *O-1* yield a much better accuracy (about 6 points better), but are an order of magnitude slower. A surprising result is the excellent performance of the rule-based *Hepple* tagger that is much faster and more accurate than any other model. This outstanding performance can be partly explained by our evaluation setting where we test on a wide range of corpora from different domains. Rule-based taggers are supposed to generalize very well and do not overfit on the training domain. It would be interesting to validate this finding on the German data, unfortunately there is no rule-based German tagger in our experiment.

On the models that are available for German, we see the same trade-off like for English, with the HunPos and the OpenNLP models being quite fast, but not as accurate as TreeTagger or Mate. Interestingly, none of the Stanford models is competitive for German.

Summarizing the overall results: *Hepple* is an excellent choice for English, while all other models for both languages suffer from a trade-off between accuracy and speed. As a consequence, researchers need to choose according to their needs. A digital humanities scholar with a couple of hundred documents to tag, may safely select the most accurate tagger, while a social media analyst looking for trends in the full Twitter stream might be better off with one of the faster alternatives.

So far, we have only considered the macro-averaged performance over all corpora. This simulates the usage scenario in which the tagger is treated as a black-box and applied to all sorts of data without caring much about the domain. In the next section, we investigate how well the models perform in different domains.

3.1 Domain-specific results

Figure 2 gives a graphical overview of the evaluation results per domain for English, while Table 3 shows the exact values. As expected, some models that are especially trained for a certain domain perform well in that domain, but not in another. One such example is the *Ark-3* model, a model specialized for social media that is among the best and fastest models on that domain, while it does not perform well on the other domains. However, there are also counter-examples like the St-6 model (trained on the WSJ) that not only performs well on formal writing, but also on the speech transcripts and social media. And of course there is the *Hepple* tagger that performs extremely well in every English domain. In general, the differences between the domains are smaller than expected. The absolute accuracy values are best for written, followed by spoken, and worst for social media which fits the expectations.

[5] In our case: Intel Core i5 2.9 GHz CPU, 16GB RAM, single core execution.

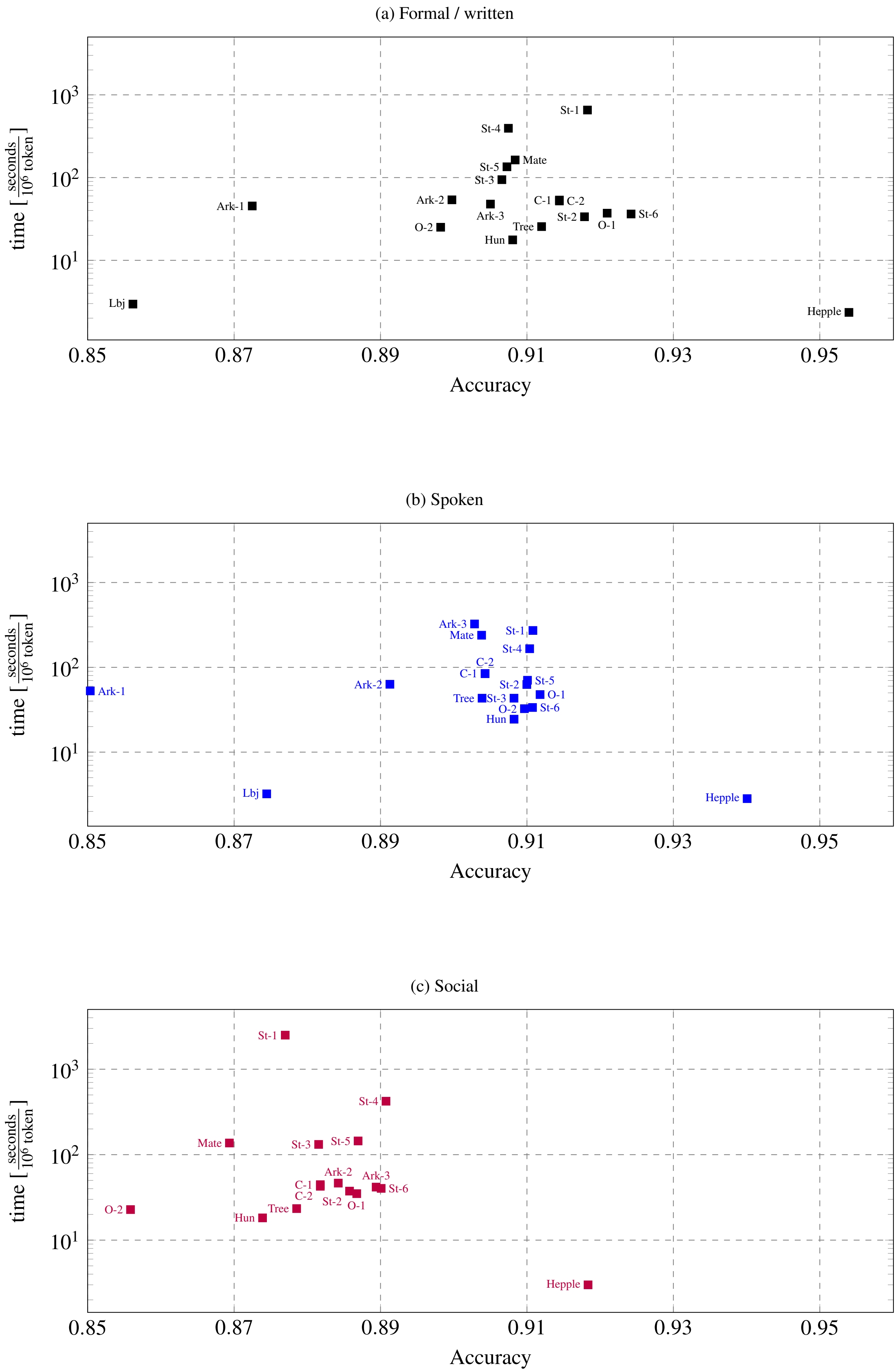
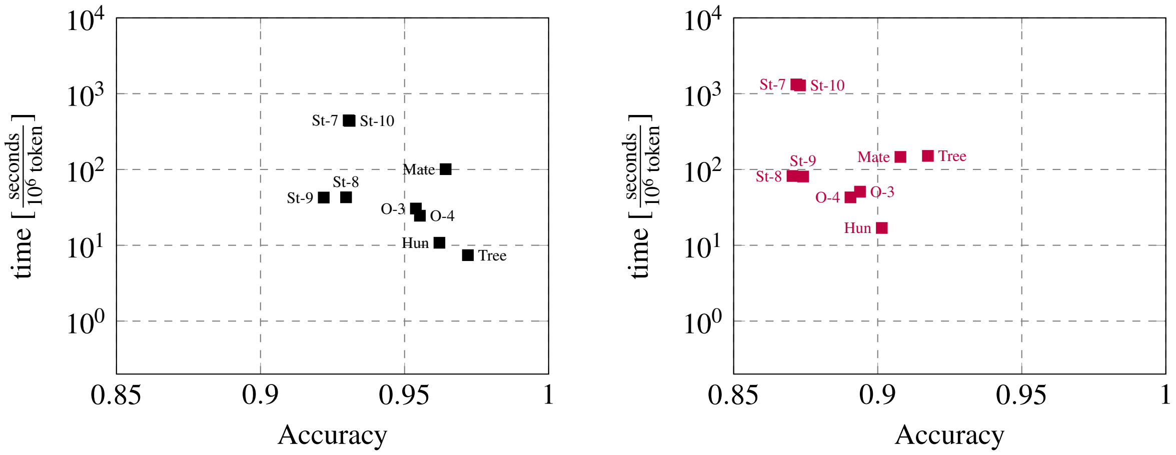


Figure 2: English results per domain.
In plot (c), *Lbj* not shown to improve readability and *Ark-1* omitted to avoid testing on training data.

	Written		Speech transcripts		Social media		Macro-Average	
	accuracy $\varnothing$ %	time $\varnothing$ ($\frac{\text{seconds}}{10^6 \text{ token}}$)	accuracy $\varnothing$ %	time $\varnothing$ ($\frac{\text{seconds}}{10^6 \text{ token}}$)	accuracy $\varnothing$ %	time $\varnothing$ ($\frac{\text{seconds}}{10^6 \text{ token}}$)	accuracy $\varnothing$	time $\varnothing$ ($\frac{\text{seconds}}{10^6 \text{ token}}$)
Ark-1	87.2	45	85.0	53			86.1	49
Ark-2	90.0	54	89.1	63	88.4	46	89.2	54
Ark-3	90.5	48	90.3	325	88.9	42	89.9	138
C-1	91.4	53	90.4	85	88.2	45	90.0	61
C-2	91.4	52	90.4	84	88.2	43	90.0	60
Hepple	**95.4**	2	**94.0**	3	**91.8**	3	**93.7**	3
Hun	90.8	18	90.8	24	87.4	18	89.7	20
Lbj	85.6	3	87.4	3	79.7	4	84.3	3
Mate	90.8	163	90.4	239	86.9	137	89.4	180
O-1	92.1	37	91.2	48	88.7	35	90.6	40
O-2	89.8	25	91.0	33	85.6	23	88.8	27
St-1	91.8	655	91.1	272	87.7	2504	90.2	1144
St-2	91.8	34	91.0	63	88.6	37	90.5	45
St-3	90.7	94	90.8	43	88.2	132	89.9	90
St-4	90.7	395	91.0	166	89.1	422	90.3	327
St-5	90.7	135	91.0	70	88.7	145	90.1	117
St-6	92.4	36	91.1	34	89.0	40	90.8	37
Tree	91.2	26	90.4	43	87.9	23	89.8	31

Table 3: English tagging accuracy and execution time. Highest accuracies per domain in bold face.



(a) Written (b) Social

Figure 3: German results per domain

	Written		Social media		Macro Average	
	accuracy $\varnothing$ %	time $\varnothing$ ($\frac{\text{seconds}}{10^6 \text{ token}}$)	accuracy $\varnothing$ %	time $\varnothing$ ($\frac{\text{seconds}}{10^6 \text{ token}}$)	accuracy $\varnothing$ %	time $\varnothing$ ($\frac{\text{seconds}}{10^6 \text{ token}}$)
Hun	96.2	11	90.1	17	93.2	14
Mate	96.4	101	90.8	146	93.6	124
O-3	95.4	31	89.4	51	92.4	41
O-4	95.5	25	89.1	43	92.3	34
St-7	93.1	445	87.2	1325	90.1	885
St-8	93.0	43	87.0	82	90.0	62
St-9	92.2	43	87.4	81	89.8	62
St-10	93.1	438	87.3	1285	90.2	861
Tree	**97.2**	7	**91.7**	151	**94.5**	79

Table 4: German tagging accuracy and execution time. Highest accuracies per domain in bold face.

When looking at the German domain-specific results (Figure 3 and Table 4), we see a similar distribution as for English with little differences between domains. An interesting exception is the *TreeTagger* that is quite fast on written data (reflecting its popularity for tagging German), but rather slow on social media. As *TreeTagger* is not open-source, we could not further investigate the reasons for this difference.

4 Conclusions and future work

In this work, we evaluated a large set of PoS tagging models on a wide range of English and German data from different domains. A surprising result is the outstanding performance of the rule-based *Hepple* tagger on English text. For German, where no rule-based tagger is readily available, we find that researchers either can choose a fast or an accurate model depending on their needs. The comprehensive results in this paper offer some guidance in this respect.

We make our full experimental framework available which will enable researchers to easily extend our analysis to other languages and taggers or compare taggers under different conditions.[6]

5 Acknowledgement

We would like to thank Richard Eckart de Castilho for his valuable input and for his incredible work on the DKPro Core framework.

References

Chris Biemann. 2006. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the COLING/ACL 2006 Student Research Workshop*, pages 7–12. Association for Computational Linguistics.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, COLING '10, pages 33–36, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

[6] https://github.com/zesch/pos-tagger-evaluation.git

Thorsten Brants. 2000. Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, pages 152–155, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jinho D. Choi and Martha Palmer. 2012. Fast and robust part-of-speech tagging using dynamic model selection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers – Volume 2*, ACL '12, pages 363–367, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 600–609, Stroudsburg, PA, USA. Association for Computational Linguistics.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. Association for Computational Linguistics.

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.

David Ferrucci and Adam Lally. 2004. UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.*, 10(3-4):327–348, September.

Eric N. Forsyth and Craig H. Martell. 2007. Lexical and discourse analysis of online chat dialog. In *Proceedings of the International Conference on Semantic Computing*, ICSC '07, pages 19–26, Washington, DC, USA. IEEE Computer Society.

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. *Proceedings of the Fifth Web as Corpus Workshop*, pages 27–35.

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 Language Technologies: Short Papers – Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA.

Sharon Goldwater and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 744–751.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 1–18, Stroudsburg, PA, USA.

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, Stroudsburg, PA, USA.

Mark Hepple. 2000. Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, Hong Kong.

Nancy Ide, Christiane Fellbaum, Collin Baker, and Rebecca Passonneau. 2010. The manually annotated sub-corpus: A community resource for and by the people. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 68–73, Stroudsburg, PA, USA.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *COMPUTATIONAL LINGUISTICS*, 19(2):313–330.

Dalos D Miguel and Rachel Edita O Roxas. 2007. Comparative Evaluation of Tagalog Part-of-Speech Taggers. *4th National Natural Language Processing Research Symposium Proceedings*, pages 74–77.

W. Nelson Francis and Henry Kuçera. 1964. Manual of information to accompany a standard corpus of present-day edited american English, for use with digital computers.

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*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Ines Rehbein. 2013. Fine-Grained POS Tagging of German Tweets. In *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.

Dan Roth and Dmitry Zelenko. 1998. Part of speech tagging using a network of linear separators. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Wojciech Skut, Hans Uszkoreit, Thorsten Brants, and Brigitte Krenn. 1998. A linguistically interpreted corpus of german newspaper text. In *Proceedings of the 10th European Summer School in Logic, Language and Information (ESSLLI'98). Workshop on Recent Advances in Corpus Annotation, August 17-28*, Saarbrücken, Germany.

Heike Telljohann, Erhard Hinrichs, Sandra Kübler, Ra Kübler, and Universität Tübingen. 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)*, pages 2229–2235.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology – Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.