Assigning Fine-grained PoS Tags based on High-precision Coarse-grained Tagging

Tobias Horsmann and 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 propose a new approach to PoS tagging where in a first step, we assign a coarse-grained tag corresponding to the main syntactic category. Based on this high-precision decision, in the second step we utilize specially trained fine-grained models with heavily reduced decision complexity. By analyzing the system under oracle conditions, we show that there is a quite large potential for significantly outperforming a competitive baseline. When we take error-propagation from the coarse-grained tagging into account, our approach is on par with the state of the art. Our approach also allows tailoring the tagger towards recognizing single word classes which are of interest e.g. for researchers searching for specific phenomena in large corpora. In a case study, we significantly outperform a standard model that also makes use of the same optimizations.

1 Introduction

When a part-of-speech (PoS) tagger assigns word class labels to tokens, it has to select from a set of possible labels whose size usually ranges from fifty to several hundred labels depending on the language. Especially for new domains or under-resourced languages, there is usually not enough training data to reliably learn all the subtle differences between a large set of labels. We thus propose to split the PoS tagging task into two steps. A first high precision step, where we only assign a coarse-grained tag which can be reliably learned also with rather limited training data, and which benefits from additional unlabeled data in the form of clusters or embeddings. And a second step, where we apply specialized tagging models that only have to choose from a much smaller set of possible labels.

Figure 1 gives an overview of our approach using, in the first step, a coarse-grained tagset (similar to the universal tagset (Petrov et al., 2012)) and in the second step the fine-grained PTB tagset (Marcus et al., 1993). In the figure, we can see how knowing the coarse-grained tag informs the second step. For example, if we already know that *beautiful* is an adjective, we only have to choose between three possible tags (JJ, JJR, or JJS) instead of 45 tags for the full PTB tagset.

Our approach requires that the coarse tagging in the first step is more accurate than using fine-grained tagging itself, as we will loose some accuracy through error propagation between the steps. We will thus first analyze how well coarse tagging can actually be done, and also focus on whether coarse-grained models transfer better between different kinds of texts, as e.g. Ritter et al. (2011) shows that a fine-grained tagger trained on newswire data doesn't work well on social media.

Creating robust and accurate coarse-grained taggers is a worthwhile task on its own, as many NLP applications actually do not require fine-grained distinctions. For example, the popular TextRank algorithm (Mihalcea and Tarau, 2004) for keyphrase extraction uses coarse grained PoS tags to build the underlying co-occurrence graph, or Benikova and Biemann (2016) use coarse-grained tags to annotate semantic relations between nominals.

Another advantage of our approach is that the second tagging step can be easily customized for specific needs, e.g. if a scholar wants to analyze the usage of a specific PoS tag. In this case, we can use fine-grained models with additional features that are informative for this sub-problem, but might not be helpful for the overall tagging task.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers,
pages 328–336, Osaka, Japan, December 11-17 2016.

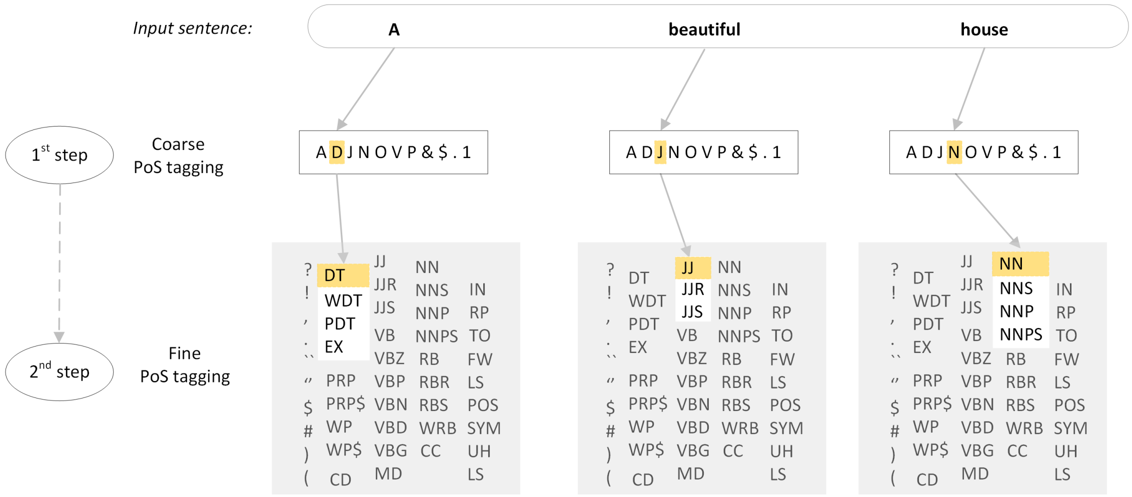


Figure 1: PoS tagging in two steps: The first step assigns a coarse tag (main word class) of a word. The second step determines the fine PoS tag based on the coarse PoS tag of the first step.

Related Work We are not aware of prior work using a similar approach. Some rule-based approaches (Brill, 1992; Hepple, 2000) assign the most probable tag to each token and then use transformation rules to correct the initial assignment. However, both steps use the same granularity and the initial assignment only reflects the prior probability but is not usable in itself.

There is also relatively little research on coarse-grained tagging. Gimpel et al. (2011) developed a tagger for Twitter data that uses a specialized coarse-grained tagset which is equivalent to our first step, but they do not aim at refining these assignments further as we do in the second step. Most approaches do fine-grained tagging first, and then map back to the universal set in order to make the different tagsets compatible. For example, Horsmann et al. (2015) use a coarse tagset to evaluate PoS tagging models using a set of corpora that are annotated with different fine-grained PoS tags.

2 Coarse-grained Tagging

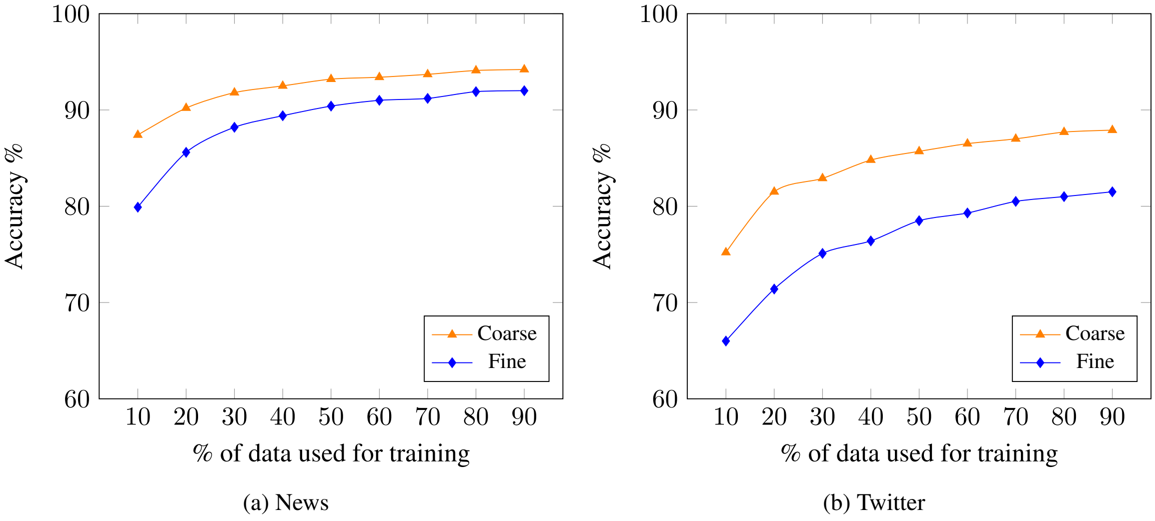
The first thing we need for our approach is a robust and highly accurate coarse PoS tagging. For domains or languages with little training data, accuracy is mainly limited by out-of-vocabulary tokens. In such situations, it seems easier to first assign a highly generalizing model that can be learned from relatively little training data instead of forcing the model to make an uninformed fine-grained decision. Thus, in this section we explore how well coarse-grained tagging actually works.

For our experiments, we use corpora from different genres (*News*, *Web*, *Chat*, and *Twitter*) in order to ensure that results are not bound to text properties only. Newswire text has a formal nature and contains few language errors, as it is usually carefully edited. Text from the web is (on average) less formal containing informal expressions and orthographic errors. Chat conversations are highly informal and often similar to spoken language, as the communication takes place between a smaller group of people with many non-standard abbreviations and orthographic errors. Twitter contains highly diverse text, as the platform is used for all kinds of purposes ranging from chat-like discussions to formal announcements. As *News* corpus, we use 46k tokens of the Wall Street Journal (WSJ) (Marcus et al., 1993), for *Web* we use 44k tokens from the GUM corpus (Zeldes, 2016) containing semi-formal text from various Wiki-platforms, as *Chat* corpus we use the NPS (Forsyth and Martell, 2007) chat corpus with 32k tokens, and for *Twitter* we use the 15k tokens Twitter messages provided by Ritter et al. (2011).

For our purposes, we need to map the fine-grained tags in these corpora to an inventory of coarse grained tags. We rely on the mappings provided by DKPro Core (Eckart de Castilho and Gurevych, 2014). We train our models using Conditional Random Fields (Lafferty et al., 2001) as implemented in FlexTag (Zesch and Horsmann, 2016) which relies on the machine learning framework DKPro TC (Daxenberger et al., 2014). As basic feature set which is common to all our models we use use ±2 tokens

	Accuracy (%)			
	News	**Web**	**Chat**	**Twitter**
fine	92.0	88.6	86.8	80.2
coarse	94.2*	92.5*	91.4*	87.2*

Table 1: Accuracy of fine tagging vs. coarse tagging. Marked values are statistically significant against the fine-grained baseline (McNemar's test, $p < 0.05$)



(a) News

(b) Twitter

Figure 2: Learning curves for 'cleanest' dataset vs. 'noisiest' dataset

as local context, the 1,000 most frequently occurring character 2-grams to 4-grams over all tokens, and whether the target token contains capitalized characters, numeric values, or special characters.

Table 1 shows the results of 10-fold cross validation on each of the four corpora. We find that coarse tagging outperforms fine tagging on all datasets (statistically significant, McNemar's test, $p < 0.05$). As expected, the size of the gain is connected to the type of corpus, ranging from 2.2 percent point on *News* to 7.0 points on *Twitter*. The improvement can almost exclusively be attributed to intra-class errors of the fine-grained model, i.e. our coarse model is differentiating between the coarse grained classes as well as the fine-grained system, but isn't forced to make an uninformed decision. In the next section, we will investigate if this gives us enough head start to improve overall performance, but before that we further investigate the properties of our coarse tagger.

2.1 Amount of required training data

A practical benefit of coarse tagging should be that the amount of required training data is considerably lower than for fine tagging, as fewer labels should need less data to be learned. We thus adapted our 10-fold cross validation experiment to train on decreasing numbers of chunks, i.e. instead of training on 9 chunks and test on the remaining chunk, we train only on 8 and test on the 10th chunk, then train on 7 and test on the 10th chunk, and so on.

Figure 2 shows the results of this learning curve experiment on *News* (our 'cleanest' corpus) and *Twitter* (the 'noisiest'). For *News*, we see the expected behavior of a larger advantage of coarse tagging for smaller amount of training data, while for *Twitter* the distance between coarse and fine tagging does not change much. Note that we have much less data for *Twitter* than for *News*, so that we could still see a similar behavior for *Twitter* if more annotated data was available.

2.2 Word class performance

So far, we have seen that coarse tagging yields in general higher accuracy than fine tagging, but it would also be interesting to see which word classes benefit the most. The possible increase in performance

	Word class	# Token	Accuracy			Word class	# Token	Accuracy	
		10^3	fine	coarse			10^3	fine	coarse
News	Adjective	0.3	77.6	78.1	Twitter	Adjective	0.7	58.6	57.7
	Adverb	1.6	84.4	84.2		Adverb	0.8	80.1	80.9
	Conjunction	1.1	99.0	98.9		Conjunction	0.3	94.3	95.6
	Determiner	4.2	98.6	98.5		Determiner	0.9	92.4	95.1
	Noun	14.3	89.9	93.5		Noun	3.5	69.1	84.1
	Numeral	1.4	96.2	95.0		Numeral	0.3	74.1	69.5
	Preposition	6.0	96.6	97.4		Pronoun	1.4	92.6	96.3
	Pronoun	1.5	98.3	98.9		Preposition	1.5	87.5	92.2
	Punctuation	6.0	99.7	99.8		Punctuation	2.0	96.5	98.7
	Verb	6.6	86.4	93.4		Verb	2.5	76.6	88.3
	Other	0.4	95.3	96.9		Other	1.5	75.4	79.9
		46.4	92.0	94.2			15.0	80.2	87.2

Table 2: Coarse tagging results by coarse word class

is linked to at least two factors: (i) the size of the remaining fine-grained label set, and (ii) how well the remaining labels can be separated. For example, the set of fine-grained labels for the noun class is rather small (four tags: NN, NNS, NNP, NNPS) and the singular/plural noun decision is rather simple in English. However, the remaining problem to differentiate between normal nouns and proper nouns is far from trivial because the distinction is often not syntactically realized, but relies on world knowledge and context. For example, *I bought an apple.* vs. *I bought an Apple.* can only be decided based on capitalization, but especially in the social media datasets this signal is not very reliable. In contrast to nouns, we do not expect adjectives to improve much, as comparative and superlative can be quite easily distinguished from the base form of the adjective and from each other.

Table 2 shows the results per coarse-grained word class, where fine-grained results are mapped to the corresponding coarse-grained value for evaluation. Again, for space reasons, we only show *News* and *Twitter* as the most extreme representatives of our evaluation datasets. As expected, nouns and verbs are responsible for most of the improvement, as they are the bigger classes and the difficult decision faced in both classes are deferred to the second step.

2.3 Cross-domain performance

So far, we have conducted all our experiments in an in-domain setting, i.e. we train and test on the same kind of data (although of course not on exactly the same data). Since for new domains there usually is no training data in the first place, it is common to fall back on models trained on the more easily available *News* corpora. In this section, we want to evaluate the difference between fine and coarse tagging in such a setting. We will train on *News* enriched with additional training data and external resources in order to create a competitive model. We then evaluate this enhanced *News* model on the remaining datasets *Web*, *Chat*, and *Twitter*.

Fine Baseline Model In order to create a strong baseline, we augment our fine-grained tagger with more training data and external knowledge. We use the same feature set as before and train the model on the WSJ sections 0-21 and additional 250k tokens taken from the Switchboard corpus. We add distributional knowledge in form of Brown clusters (Brown et al., 1992) that we trained on 100 million tokens of English tweets crawled between 2011 and 2016. When we evaluate on the WSJ sections 22-24, we achieve an accuracy of 96.4% which is on par with the state of the art for fine-grained tagging which ranges from 96.5% (Brants, 2000) to 97.6% (Huang et al., 2015) according to the ACL wiki.[1]

Coarse Model Our coarse model is also trained on the WSJ section 0-21. An advantage of using coarse tags is the availability of many annotated corpora that can be easily mapped to coarse tags regardless of

[1] http://aclweb.org/aclwiki/index.php?title=POS_Tagging_(State_of_the_art)

	Accuracy (%)		
	News → Web	**News → Chat**	**News → Twitter**
fine	92.3	73.6	80.8
coarse	96.0*	88.9*	90.6*

Table 3: Accuracy of a fine trained and coarse trained model for tagging across domains. Marked values are statistically significant against the baseline (McNemar's test, $p < 0.05$)

the actually used fine tags. This immensely extends the pool of annotated training data that we can choose from. We thus additionally add 250k tokens of the Switchboard corpus as well as a PoS dictionary feature based on the three most frequent coarse tags of each word derived from the British National corpus (Clear, 1993). Furthermore, we add additional annotated Twitter data (Owoputi et al., 2013) to inform our classifier about phenomenons found in the social media domain. Please note that neither of these extension is possible for the baseline model because the tagsets are not compatible on the fine-grained level. However, we can use those resources for coarse grained tagging by mapping the fine tags to the same coarse tagset.

Table 3 shows the results of tagging the *Web*, *Chat*, and *Twitter* datasets, while we exclude *News* as we have been using it for training the models.[2] We find that in this cross-domain setting, our coarse-grained tagger significantly outperforms the fine-grained version for all three datasets. Especially the results on the *Chat* dataset are informative, as the news-trained model can obviously not be transferred well to chat data. In contrast, our coarse-grained version performs even better on *Chat* than on *Twitter* showing that it generalizes well.

3 Fine-grained Tagging

As we have shown in the previous section, we can assign coarse tags with higher accuracy than fine tags. This gives us the necessary head start for our second tagging step, as errors from the first stage will be propagated into the second stage and cannot be corrected anymore. For example, if we wrongly assigned the coarse tag `N` to a verb in the first step, we will in the second step apply the specialized classifier for nouns that has no chance of assigning a verb tag.

We implement the fine tagging by using a dedicated model which assigns fine PoS tags belonging to one coarse word class. For this step, we use a Support Vector Machine (Joachims, 1998) as implemented in Weka (Holmes et al., 1994), because this sub-task is not easily implemented as a sequence classification task. We use the same feature set as before and train the second step models on the WSJ considering only words belonging to the same coarse word class. Two coarse word classes (conjunction and numbers) do not have further fine-grained specializations in the PTB tagset, i.e. they can be mapped one-to-one to their fine PoS tag.

In order to assess the full potential of our approach, we will first ignore error propagation between the two tagging stages. This means, we assume an oracle condition that assigns correct coarse-grained tags. Afterwards, we will evaluate our full model with error propagation and check against the hypothetical performance of the oracle.

We show the results of fine tagging under oracle condition in Table 4 in comparison to our fine baseline. Marked values show statistically significant improvements against the baseline (McNemar's test, $p < 0.05$). Under oracle condition our model performs significantly better on all our for evaluation corpora. In the last row of Table 4, we show the results of our two-step approach, where errors made in the coarse step are propagated to the fine-grained tagging. Our results are on par with the state-of-the-art fine baseline for three out of four datasets, and are significantly better for the *Chat* corpus.

In order to better understand the influence of coarse errors, we simulate a certain level of error propa-

[2] Dataset with additional tags not occurring in the training data pose special problems. We treat the few unknown tags as always correct in order not to influence the relative difference between the fine and coarse setup too much.

	Accuracy (%)			
	News	**Web**	**Chat**	**Twitter**
Fine baseline	96.4	92.3	73.6	80.8
Oracle 100%	98.4*	95.8*	83.0*	89.3*
Two-step	96.1	92.4	75.8*	81.9

Table 4: Accuracy of fine-grained tagging using oracle coarse-grained tagging and two-step tagging. Marked values are statistically significant against the baseline (McNemar's test, $p < 0.05$)

gation by using an oracle that only returns the correct tag with a certain probability (uniformly sampled over all tags). Figure 3 shows the resulting relationship between a certain level of coarse-grained accuracy and the resulting fine-grained accuracy. We see that both are in a similar linear relationship for all datasets. Measuring the slope, we can approximate that a 1 percent point improvement in coarse accuracy translates into an improvement between .8 and .9 percent points in fine-grained accuracy. The accuracy of our two-step tagging is marked by a black dot. Its location is always on or very close to the hypothetical performance of the oracle. We can thus conclude that our predictions are quite accurate and better coarse performance will really lead to better fine performance. On *Chat* our approach performs slightly better than predicted, which we take as an indicator that our error propagation experiment is a rather conservative lower-bound for the impact of coarse tagging errors.

4 Tag-specific Optimization

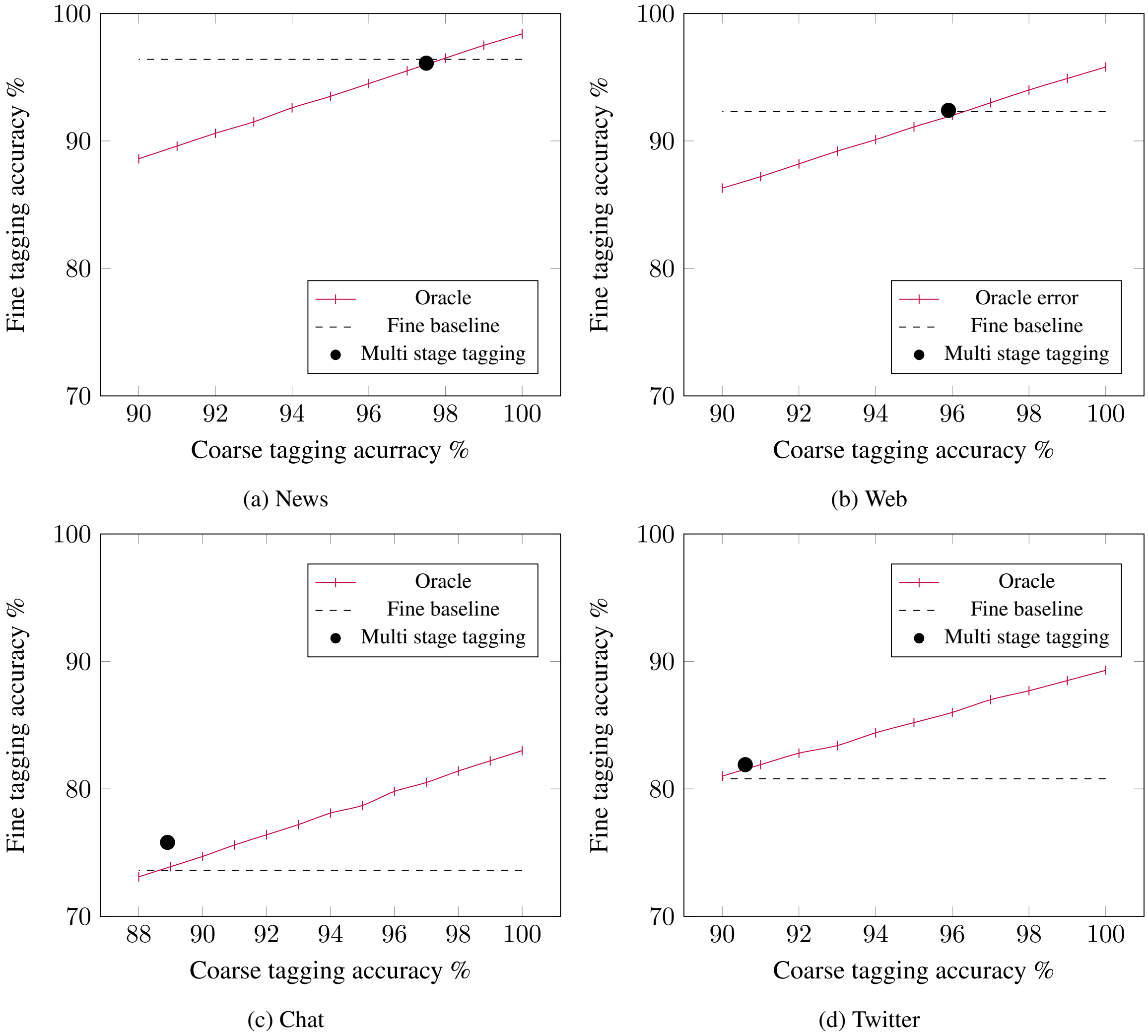
While it is hard to optimize the tagging accuracy for a specific fine-grained tag using traditional taggers, our approach offers a straightforward way to train a second stage model that is tailored towards such a task. For example, a scholar might be only interested in finding past participle verbs (word class `VBN` in the PTB tagset). However, they are easily confused with past tense verbs (word class `VBD`) due to their identical word surface form as in the example below.

> *Mr. Smith was arrested/VBN and* ***charged/VBN*** *along with the others when he returned to Namibia this month*

A tagger is easily mislead to tag the second VBN (bold faced) as past tense verb rather than past participle, because the auxiliary is outside a narrowly defined ± 2 context window. Note that the past tense decision might even be correct in a very similar context like *Mr. Smith was arrested/VBN and* ***went/VBD*** *along anyway.* So the tagger either needs to learn the compositional semantics of 'charged along' from a large number of annotated examples or by using additional external knowledge. We thus argue that for our specialized `VBN` classifier it is easier than in a global setting to utilize a wider context window and use more lexical features.

If we wish to optimize the detection accuracy for VBN in our approach, we can ignore the performance on the other tags. Thus, we transform the problem into a one vs. all classification, i.e. we further reduce the decision complexity in the second tagging step to a binary decision between `VBN` and ¬`VBN`. In order to avoid a class bias, we ensure a balanced distribution of both classes in the training data and train our model again on the WSJ. We extend the feature set to provide additional hints for the classifier that one of the words outside of the ± 2 context window is an inflected form of *have*, *be*, or a modal verb.

Table 5 shows the results of our optimization experiment. We evaluated the accuracy of tagging only the word class VBN. We report accuracy for the fine baseline and our two-step model in (i) standard configuration and (ii) the `VBN` optimized version. The first thing to notice is that without optimization our standard two-step model performs worse than the fine-grained baseline. However, our optimized two-step model significantly outperforms the fine-grained baseline on all datasets. This shows that our separation of the tagging process in two steps allows for an easy way of customizing a PoS tagger towards specific needs by incorporating linguistically motivated features.



(a) News

(b) Web

(c) Chat

(d) Twitter

Figure 3: Relationship between fine-grained accuracy and error-level of the coarse-grained tagging. The horizontal dashed line shows the baseline performance and the black dot shows the accuracy of our two-step PoS tagging.

	VBN Accuracy (%)							
	News		**Web**		**Chat**		**Twitter**	
	w/o	w	w/o	w	w/o	w	w/o	w
fine	86.7	87.4	81.2	82.4	75.7	77.0	70.0	76.4
two-step	84.6	90.7*	79.0	86.7*	74.0	81.3*	64.3	81.4*

Table 5: Results for detecting **VBN** with and without optimization. Marked values are statistically significant against the fine-grained baseline (McNemar's test, $p < 0.05$)

5 Conclusions

We introduce a new two-step PoS tagging approach, where first a coarse-grained tag is assigned with high precision, and in a second step a specialized fine-grained classification with heavily reduced decision complexity is applied. We show that the accuracy of coarse-grained tagging is significantly higher when directly learning a coarse model compared with a fine-grained model. This especially holds in a cross-domain setting where coarse models generalize much better than fine models. As many NLP applications rely on coarse-grained tags, this finding has a huge potential for practical improvements in downstream tasks. If we evaluate the fine-grained accuracy, we find that our two-step approach performs on par with the classical single-step paradigm, except for the especially challenging chat messages where it is significantly better. Using an oracle for coarse-grained tagging, we show that our approach has a huge potential for improving the overall accuracy, as improving coarse-grained accuracy translates almost directly into coarse-grained accuracy. In the common setting that a researcher is only interested in the performance of a specific fine-grained tag, we show that our approach can be easily optimized and then significantly outperforms the classical approach that does not improve even if the same optimizations are used.

In future work, we aim to further improve coarse grained tagging, as our analysis shows that coarse-grained improvements nearly linearly translate into fine-grained improvements. We also aim at further improving fine-grained tagging, as we see large potential for further specializing the feature sets used for different word classes.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant No. GRK 2167, Research Training Group "User-Centred Social Media".

References

Darina Benikova and Chris Biemann. 2016. SemRelData Multilingual Contextual Annotation of Semantic Relations between Nominals: Dataset and Guidelines. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4154–4161, Portoroz, Slovenia.

Thorsten Brants. 2000. TnT: A Statistical Part-of-speech Tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 224–231, Seattle, Washington.

Eric Brill. 1992. A Simple Rule-based Part of Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy.

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.

Jeremy H. Clear. 1993. The British National Corpus. pages 163–187. Cambridge, MA, USA.

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.

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*, pages 19–26, Washington, DC, USA.

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*, pages 42–47, Portland, Oregon.

Mark Hepple. 2000. Independence and Commitment: Assumptions for Rapid Training and Execution of Rule-based POS Taggers. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 278–277, Hong Kong.

Geoffrey Holmes, Andrew Donkin, and Ian H. Witten. 1994. *WEKA: A Machine Learning Workbench*. Working paper series (University of Waikato. Department of Computer Science).

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)*, pages 22–30, Essen, Germany.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *ArXiv e-prints 1508.01991.*

Thorsten Joachims, 1998. *Text categorization with Support Vector Machines: Learning with many relevant features*, pages 137–142.

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 8th International Conference on Machine Learning*, pages 282–289, San Francisco, CA, 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.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain.

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 *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, Georgia.

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 2012)*, pages 2089–2094, Istanbul, Turkey.

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*, pages 1524–1534, Edinburgh, Scotland.

Amir Zeldes. 2016. The GUM corpus: creating multilayer resources in the classroom. *Language Resources and Evaluation*, pages 1–32.

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.

DOI: 10.17185/duepublico/72094
URN: urn:nbn:de:hbz:464-20211021-150126-2

Horsmann, T., Zesch, T (2016). Assigning Fine-grained PoS Tags based on High-precision Coarse-grained Tagging. In: *Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers,* pp. 328-336. The COLING 2016 Organizing Committee. https://www.aclweb.org/anthology/C16-1032