Variations on the theme of variation: Dealing with spelling variation for fine-grained POS tagging of historical texts

Fabian Barteld[a,b,c] Chris Biemann[b] Heike Zinsmeister[a]

[a]Institut für Germanistik, Universität Hamburg
[b]Language Technology Group; Department of Informatics, Universität Hamburg
[c]Department of Linguistics, Ruhr-University Bochum

firstname.lastname@uni-hamburg.de

Abstract

In this paper, we present experiments on POS tagging historical texts that contain spelling variation. The experiments are conducted in a low-resource scenario with a small amount of training data (here: 12,000 tokens). We investigate different ways of dealing with spelling variation in such a situation on different variants of historical German. Firstly, we add character n-grams as features to the tagger to enable it to learn spelling variation. Our tagging experiments show that this improves accuracy when there is enough variation in the data, but leads to a decrease in accuracy if the amount of variation is low. Secondly, we preprocess the data before training and applying the tagger, reducing spelling variation by normalization, rule-based simplification and substitution of spelling variants. All three methods improve tagging accuracy in comparable levels. Since normalization has the drawback of requiring additional resources, we recommend rule-based simplification and substitution of spelling variants for low-resourced settings. Finally, we evaluate the utility of additional unlabeled data to create word embeddings and employing external resources, which we use to further improve tagging accuracy.

1 Introduction

When developing a part-of-speech (POS) tagger for historical texts, one has to deal with multiple problems that are not encountered with standard texts. One of these problems is the low-resource nature of historical texts: Annotated training data is sparse and also unlabeled data for supporting domain adaption methods and semi-supervised approaches is not as readily available as for most contemporary languages. Furthermore, historical texts do not have a fixed orthography and therefore exhibit spelling variation, which means that words are spelled differently across time and region—and often even within one and the same text. To address these issues, many approaches perform normalization, i.e. a mapping of word tokens to a contemporary standard form, for the purposes of reducing the variation and for making available language resources applicable. For German, work in that direction is mainly concerned with Early New High German (1350–1650) or newer texts.

In this paper, we perform experiments on historic variants of German, that are either older than Early New High German—Middle High German (1050–1350)—or historical variants of Low German, a German dialect—Middle Low German (1250–1650). These texts differ substantially more from contemporary Standard German than Early New High German. Therefore, we do not aim to apply a POS tagger developed for contemporary German to the data after normalization. Instead, we train a POS tagger directly on the data, limiting ourselves to a set of about 12,000 tokens for training. In order to overcome the problems that spelling variation poses to statistical tagging, we experiment with different techniques: We compare the effects of using an adapted feature set for the tagger, reducing the spelling variation before tagging and making use of additional information from external resources.

2 Related work

Techniques for dealing with the problems that historical texts pose for POS tagging, in particular spelling variation, can roughly be categorized into three categories:

1) Development of specialized taggers. One approach is to develop a specialized part-of-speech tagger that deals with spelling variation directly, e.g. by including specific features. One example for this is Koleva et al. (2017)'s solution, who present experiments on Middle Low German texts with a memory-based learner and a conditional random field tagger with differing sets of features including prefix and suffix n-grams. The authors conclude that a tagger using such features handles spelling variation itself, while normalization only leads to marginal improvements for tagging accuracy.

In Section 5, we present experiments with adding additional features to a part-of-speech tagger.

2) Reduction of spelling variation. Another approach is to deal with spelling variation in a preprocessing step and apply a generic part-of-speech tagger afterwards. Dipper (2010) performs tagging experiments on a corpus of Middle High German texts in three different versions: A strict *transcription* that captures many peculiarities of the script, for example superposed characters, a *simplified* version, where most of these peculiarities are removed, e.g. superposed characters are brought into sequence, and a *normalized* version where spelling variants are reduced by mapping the words to an artificial Middle High German standard that is traditionally used by philologists. Dipper's experiments show that training and tagging lead to better results with less variable variants: Normalized data is better than simplified data, which in turn is better than using the strict transcription.

One limitation here is that the normalization of the Middle High German texts has been done only semi-automatically and—to our knowledge—there is no previous work that explores the utility of automatic normalization in the sense of mapping words to a standard form when training a tagger except the work of van der Goot et al. (2017), who present experiments on English tweets. They come to the conclusion that while normalization improves tagging accuracy, using word representations obtained from a large amount of unlabeled data gives lager improvements. Combining both only leads to small further improvements.

Logačev et al. (2014) as well as Barteld et al. (2015) aim at reducing spelling variation by detecting likely pairs of spelling variants and substituting unknown words with a known word that is a spelling variant. Another way to reduce spelling variation is rule-based simplification, employed e.g. by Adesam and Bouma (2016) for POS and morphological tagging.

In Section 6, we present experiments with rule-based simplification, normalization and spelling-variant detection for reducing variation.

3) Usage of external resources. The usage of external resources mainly aims at overcoming the lack of training data. In addition to preprocessing the historical data before training and applying a part-of-speech tagger as described above, normalization of non-canonical texts can be—and is usually—used to make them accessible for standard language tools by mapping historic word forms to their contemporary cognates in order to apply the respective tools—a tagger or other resources developed for contemporary texts—to the historical data, achieving reasonable results (Bollmann, 2013; Tjong Kim Sang et al., 2017).

As normalization cannot deal with all deviations of historical texts from their modern equivalents that have an effect on POS tagging such as syntactic changes, there are also experiments to combine normalization with domain adaptation (Yang and Eisenstein, 2016).

In Section 7, we present experiments with using normalization in order to apply an existing tagger to the Middle High German data and additional unlabeled data for training word embeddings.

We cannot compare our work to the results of these studies directly as they use different, mostly unpublished datasets.

3 Data

Publicly available annotated corpora for historical German that allow for experiments on different techniques for POS tagging have become available only recently.

We use texts from two different historical variants of German: Middle High German (1050–1350) and Middle Low German (1200–1650). The texts are extracted from the Reference Corpus Middle High German (*ReM*) (Klein et al., 2016) and the Reference Corpus Middle Low German (*ReN*) (ReN-Team, 2018). ReM consists of 394 texts with 2,448,379 tokens annotated with part of speech (POS), morphology, lemma and a normalized Middle High German form. At the time of writing, ReN is still work in progress. The data for our experiments comes from its pre-release 0.6 consisting of 50 texts with 339,664 tokens annotated with POS, morphology and lemma.

While these two corpora allow for training taggers for Middle High and Middle Low German with a good amount of training data, for researchers working with historical texts from other periods or specific genres, training data is sparse. We want to simulate such a true low-resource setting in our experiments and, therefore, limit ourselves to a selection of six texts from each of the corpora and train the taggers on only about 12,000 tokens as Schulz (2018) shows for Middle High German that the learning curve flattens after 12,000 tokens. After that, adding 2,000 tokens more as training data only leads to small improvements below 1%. This suggests that 12,000 tokens would be a good start in a setting where training data for a tagger needs to be generated.

ReM and ReN consist of texts from different points in time and different dialect areas. For our selection, we pick texts that come from similar points in time and dialects to minimize the amount of spelling variation that is due to temporal and dialectal differences.

ReM consists of different subcorpora. We limit our selection to texts from its MiGraKo subcorpus and use only prose texts from the upper German dialect area from the first half of the 13th century. This selection leads to six texts.

For ReN, we limit the data to texts from the 14th century, taking four texts from Northern Low Saxon and two texts from Eastphalian.

In both corpora, the texts have been manually tokenized and split into sentences. We use these segmentations for our experiments. As training data, we use roughly the first 2,000 tokens (always using complete sentences) from each text of both datasets. This simulates the approach where a POS tagger for a low-resourced language is created by annotating the beginnings of texts used for training a model that automatically annotates the remaining parts of the texts. For development, we use the following 1,000 tokens and for testing the next 1,000 tokens (again, complete sentences).

The corpora present tokens in three different versions, which we exemplify with the word *biſt* (‘(you) are’)[1]:

1. **transcription**: This version uses specific markup developed in the projects and encodes a lot of the peculiarities of the script, e.g. *(b*)i$t|, where *(X*) indicates the usage of an initial, $ encodes a long s (ſ) which was used nearly interchangeably with the graph *s* and | encodes that there is no space after this word.

2. **strict**: This version encodes many of the peculiarities using unicode, but does abstract away from features of the text such as initials, e.g. *biſt*.

3. **simple**: This version removes variation by mapping non-ASCII characters to their ASCII counterparts, e.g. the long s (ſ) to s, as in *bist*. This version has been created with rule-based mappings.

For our experiments, we use the *strict* and the *simple* version. The *strict* version captures a lot of the spelling variation in the texts while not using project-specific markup, so models trained on this version will be more useful with other resources. The *simple* version is used for experiments on reducing spelling variation with rule-based simplifications. With ReN, *simple* does not deviate much from *strict*. Here we use the rules presented in Koleva et al. (2017) to prepare a simplified version. Capitalization is ignored in all experiments.

Table 1 shows statistics on the datasets. The POS and morphological annotations are done with tagsets derived from the Standard German STTS (Schiller et al., 1999), namely HiTS (Dipper et al., 2013) for ReM, and HiNTS (Barteld et al., 2018) in the case of ReN. These tagsets are very fine-grained. One specificity introduced in HiTS is the usage of two types of POS tags: a context-specific tag and a lexeme-specific tag. For our experiments we use the concatenation of both tags as POS tag. Spelling variation is measured by giving the proportion of morphological words that are realized by more than one type in the data (Barteld, 2017).

	ReM	ReN
Size of training set	12,108	12,025
Size of development set	6,064	6,024
Size of test set	6,062	5,667
Number of tags in training set	79	70
Training set variation (strict)	22.81%	18.11%
Training set variation (simple)	18.12%	16.81%

Table 1: Dataset statistics.

4 Baselines

We train available taggers on the data to establish some baselines: RFTagger, an HMM tagger using

[1] They are called *trans*, *utf* and *ascii* in the CorA-XML format (`https://cora.readthedocs.io/en/latest/document-model/\#token-representations`).

decision trees to estimate the probabilities (Schmid and Laws, 2008), HunPos (Halácsy et al., 2007), a re-implementation of TnT (Brants, 2000) an HMM tagger using trigrams and Marmot, a CRF tagger (Müller et al., 2013). The taggers are trained with standard settings.[2]

Table 2 shows the results for the tagger on the *strict* version of the datasets.[3] Marmot leads to the best results across both datasets. Significant improvements over the tagger below are marked with ‘*’.[4]

Tagger	ReM	ReN
Marmot	**84.05***	**85.44**
HunPos	82.32	84.78*
RFTagger	81.68	83.95

Table 2: Baseline tagging accuracies (development). ‘*’ marks a significant improvement over the tagger below.

In the following sections we look into how to improve these results with 1) additional features, 2) reduction of spelling variation and 3) the usage of external resources.

5 Features for dealing with spelling variation

Spelling variation increases the risk for a tagger to encounter unknown words, however spelling variants themselves will show a large character overlap. Therefore, taking subword information into account seems promising for tagging historical texts, as Dipper (2010) pointed out. In this section, we present experiments on tagging the *strict* version of the texts using character n-gram features.

All of the baseline taggers already use character n-gram information in some way. Both RFTagger and HunPos use suffix information to estimate tag probabilities for unknown words, the maximum length of the suffixes is set to 7 (RFTagger) and 10 (HunPos). Marmot uses prefixes and suffixes up to a given length as features for rare words, length 10 in the standard settings. To get an insight about the impact of subword information, we set the maximum affix length for Marmot to $\{4, 7, 10, 13, 16\}$.

Next to prefix and suffix features, Marmot also allows to use infix features, which is disabled by default. The length of the infixes is also governed by the maximum length parameter. The results for the different lengths with and without the usage of infixes are given in Table 3.

Max. length	Infix	ReM	ReN
16	+	84.83*	85.91
13	+	84.83*	85.91
10	+	84.84*	**85.94**
7	+	84.88*	85.86
4	+	**84.93***	85.89
16	-	84.07	**85.51**
13	-	84.12	**85.51**
10	-	84.05	85.44
7	-	**84.15**	85.38
4	-	83.94	84.88#

Table 3: Character n-gram features (development). ‘*’ marks a significant improvement over the standard settings (Max. length 10, without infixes), ‘#’ marks a significant loss in performance.

While the best settings differ for the datasets, there are two general points: 1) Without infix features, the numbers show that the increase from 4 to 7 leads to a high increase in accuracy while higher affix lengths only change the accuracy marginally, so the default value of 10 is a reasonable choice for our datasets as well. 2) Adding infix features leads to improvements for both datasets, however they are only significant in the case of ReM. The maximum length of character n-grams does not lead to significant differences in the accuracy when using infix feature. We use length 4 for further experiments.

In the default settings of Marmot, rare words are defined as words with a training data frequency of up to 10. For Modern German, Marmot has been trained on the first 40,474 sentences of the TIGER treebank (Müller et al., 2013). This is a substantially larger dataset than the 12,000 tokens used here. Hence, using the same frequency threshold leads to an effectively lower threshold for rare words. Still, it might be useful to include character n-grams for more words in order to enable the tagger to learn spelling variation. We experiment with setting the maximum frequency for rare words to $\{5, 10, 15, 20, 25, 30, 35, 40, \infty\}$. The results in Table 4 show that 10 is a reasonable default: Higher thresholds up to 30 seem to give better re-

[2] HunPos crashes if no token consisting only of digits exists in the training data. Therefore, we added a dummy token to the ReM data.

[3] Tagging results are given as accuracies in percentage points.

[4] For all tests in this paper we use McNemar’s test (McNemar, 1947) with continuity correction (Edwards, 1948) and a significance level of 0.05.

sults but the improvements are not significant. We set the rare words frequency to 30 for further experiments as this gives the best results for both datasets. Adding the features to all words (∞) does not improve the tagging accuracy.

Freq.	ReM	ReN
∞	84.81	86.09
40	84.89	85.91
35	84.93	86.06
30	**85.06**	**86.16**
25	85.03	86.12
20	84.86	86.04
15	85.04	**86.16**
10	84.93	85.89
5	84.96	85.61

Table 4: Different frequency thresholds for rare words (development).

To conclude these experiments: Tagging accuracy for both datasets can be improved by adding infix features and using higher frequency thresholds for rare words than for Modern German. The utility of these features seems to depend on the amount of spelling variation in the data as the differences are higher for ReM, which has a higher proportion of variation. In the remainder of the paper, we call Marmot with the original feature set **Marmot-*orig*** and with the tweaked feature set—using prefixes, suffixes and infixes of length up to 4 for words with a frequency up to 30 in the training data—**Marmot-*hist***. Table 5 gives a comparison of both on the test sets. While Marmot-*hist* is significantly better than Marmot-*orig* for ReM, for ReN Marmot-*orig* is better than Marmot-*hist*, however, the difference is not significant.

Tagger	ReM	ReN
Marmot-hist	**85.86***	83.71
Marmot-orig	84.28	**84.15**

Table 5: Tagging results for Marmot with original feature set (Marmot-*orig*) and a feature set tweaked for historical texts with spelling variation (Marmot-*hist*) (test). '*' marks a significant improvement of Marmot-*hist* over Marmot-*orig*.

6 Reducing spelling variation

In the previous section, we have looked into character n-gram features to enable the tagger to better deal with spelling variation. An alternative approach is to preprocess the texts and remove spelling variation before applying the tagger. In this section, we look into different ways to achieve this and their interaction with the enhanced feature set of Marmot-*hist*.

A simple way to reduce spelling variation is to design a set of rewrite rules in order to conflate spelling variants. One example for Middle High German would be to substitute the long s (ſ) with a regular round s, removing the variation between these two characters. For experiments with this approach, we use the *simple* version of the texts. Table 6 contains the tagging accuracy when training and tagging on *simple*. For ReM the accuracy improves significantly by nearly 1% with Marmot-*hist*. Marmot-*orig* even improves further, rendering the differences between both settings as insignificant, indicating that the infix features actually capture spelling variation and are less useful in situations with less variation. For ReN, simplification only improves the tagging results for Marmot-*orig*.

Tagger	ReM	ReN
Marmot-hist	**85.98***	**86.12**
Marmot-orig	85.55	85.81

Table 6: Tagging accuracy with simplification (development). '*' marks a significant improvement over tagging the strict version with Marmot-*hist*.

To further investigate the impact of using infix features, we again experiment with setting the maximum frequency for rare words to $\{5, 10, 15, 20, 25, 30, 35, 40, \infty\}$. Table 7 shows that infix features help to improve the tagging accuracy, however with less variation it is better to add them to fewer words: For ReN, thresholds of 35 and 40 show a significant drop in performance compared to a threshold of 10.

Dipper (2010) has already shown that making use of normalization leads to even further improvements regarding Middle High German. Normalization abstracts away from dialectal differences. An example for this is the Middle High German *maiſters* ('master'), its simplified version is *maisters* with the long s changed to a round s. Its normalized version again is *meisters*, which abstracts away from a general variation between *ai* and *ei* in Middle High German. In the semi-automatically normalized version of the ReM training data[5]—there

[5] For some types, e.g. punctuation, no normalization is given. In this case, we use the simple version.

Freq.	ReM	ReN
∞	85.85	86.29
40	85.82	85.92#
35	85.92	85.99#
30	85.98	86.12
25	85.78	86.14
20	85.95	86.19
15	85.95	86.30
10	**86.02**	**86.49**
5	85.92	86.14

Table 7: Different frequency thresholds for rare words with simplification (development). '#' marks a significant loss in performance compared to the max.frequency setting of 10.

is no normalized version of ReN available—only 5.56% of the morphological words are realized by more than one type, which is a substantial reduction of spelling variation (cf. Table 1).

For experiments on automatic normalization of the ReM texts we use cSMTiser,[6] a normalization tool using character-level machine translation that was one of the best performing systems in the CLIN27 Shared Task (Tjong Kim Sang et al., 2017). The techniques have been described in Ljubešić et al. (2016) and Scherrer and Ljubešić (2016). Using cSMTiser, we train a normalization model on the normalization of the training set using only tokens. The model normalized 86.23% tokens correctly on the development set.[7]

Tagger	Normalization	ReM
Marmot-hist	gold	89.51*
Marmot-orig		**89.71***
Marmot-hist	automatic	85.08
Marmot-orig		**85.39**

Table 8: Normalized Middle High German (development). '*' marks a significant improvement over tagging the strict version with Marmot-*hist*.

Table 8 shows that tagging accuracy is significantly higher when tagging the gold normalized version than the strict version for both feature sets. With automatic normalization the improvement in tagging accuracy is lower. For the automatically normalized data—as well as for the gold normalization—using Marmot-*orig* leads to better results than using Marmot-*hist*. Although the difference is not significant this shows again that the feature engineering was tailored to texts with spelling variation.

As an alternative to normalization, we experiment with spelling variant detection and substituting out-of-vocabulary (OOV) words with their known spelling variant if possible (Barteld et al., 2015). To get an impression on how much improvement is possible with this technique, we calculate upper bounds by substituting OOV words with the most frequent spelling variant from the training data if one exists. Spelling variants are defined by having the same POS tag, morphology and lemma (spellvar). As not only correct substitutions will help the tagger but also substitutions with another known word that just has the same POS or distribution might help (Barteld et al., 2015; Kolachina et al., 2017), we also substitute OOV words with the most frequent known type that has the same POS (spellvar$_{\text{pos}}$). Table 9 shows the results for these upper bounds. In contrast to the experiments with normalization above, this time Marmot-*hist* performs better than Marmot-*orig*. This can be explained by the fact that the variation is not reduced in the training data. With spelling variant substitution, we achieve an improvement that is larger than the improvement obtained with automatic normalization but lower than the upper-bound improvement with normalization. Applying the not-so-strict definition of spelling variation, leads to substantial gains, which we attribute to the fact that this excludes unseen words in the task of POS tagging.

For automatic spelling variant detection, we apply a variant of the approach proposed in Barteld (2017): For all unknown types in the development data, we select all known types with a Levenshtein distance (Levenshtein, 1966) of 1 as candidates and filter this set using supervised machine learning. In contrast to the work described in Barteld (2017), we do not apply subsampling to the training data and do not use a frequency threshold. Instead, we train a bagging classifier as this addresses both, the imbalanced data and the fact that training data might contain examples that are falsely labeled as negative (PU-learning) (Galar et al., 2012; Mordelet and Vert, 2014). As a base classifier, we use a SVM as proposed by these works. Our implementation is available at `https://github.com/fab-bar/SpellvarDetection`. It uses the Python libraries Scikit-learn (Pedregosa et al.,

[6] `https://github.com/clarinsi/csmtiser`

[7] Training a model to normalize whole sentencens, thereby including token context into the normalization, led to worse results with the small amount of training data.

Tagger	ReM spellvar	 spellvar$_{pos}$	ReN spellvar	 spellvar$_{pos}$
Marmot-hist	86.71*	93.27*	87.30*	91.88*
Marmot-orig	86.20*	92.71*	85.67	91.50*

Table 9: Spelling variant substitution – upper bounds (development). '*' marks a significant improvement over tagging the strict version with Marmot-*hist*.

Tagger	ReM spellvar	 spellvar$_{norm}$	 spellvar$_{pos}$	ReN spellvar	 spellvar$_{pos}$
Marmot-hist	**85.69***	**85.69***	85.46*	86.37	**86.39**
Marmot-orig	84.78	84.76	84.71	85.91	85.62

Table 10: Spelling variant substitution with automatic spelling variant detection (development). '*' marks a significant improvement over tagging the strict version with Marmot-*hist*.

2011) and Imbalanced-learn (Lemaître et al., 2017). We only use aligned character n-gram features for the SVM. From the spelling variants identified by this method, the most frequent type (measured on the training data) is chosen.

Training pairs for the spelling variant detection can be obtained in different ways. We test three settings: 1) using lemma, POS and morphology (spellvar), 2) using the normalization (spellvar$_{norm}$) and 3) using only POS (spellvar$_{pos}$). While 1) and 2) will lead to reliable training data, option 3) leads to more noisy training data, however this option is especially interesting in the low-resource settings as no additional annotation or data is needed.

Table 10 shows that substituting automatically detected spelling variants for OOV words results in improvements of the tagging accuracy that are comparable to the improvements obtained with automatic normalization. As with the upper-bound experiments, Marmot-*hist* gets better results. Using only the POS annotation gives results that are only slightly worse than the ones with more reliable training data. In the case of ReN, they are actually even slightly better than those.

Table 11 shows a comparison of automatic normalization, simplification and spelling variant substitution trained with lemma, POS and morphology and only POS on the test set. All of the methods for spelling reduction improve the respective tagger. For ReM, the combination of Marmot-*hist* and simplification leads to the best results. For ReN, this combination leads to the second best result with spelling variant detection using lemma, POS and morphology for the training data leading to the best results.

7 External resources

In the previous sections, we have limited ourselves to using about 12,000 tokens as training data, for some experiments exploiting annotations like normalization or lemma. In this section, we experiment with using other resources in addition to the training data. These fall into two categories: Word representations from additional unlabeled data and—in the case of ReM—an existing tagger.

For additional unlabeled data, we use the texts from the corpora that are not used in the POS tagging experiments. In the case of ReM these are 392 texts, 2,437,090 tokens, in the case of ReN only 44 texts, 259,192 tokens. We try three different ways to obtain word representations from these datasets: PPMI-SVD, Skip-gram with negative sampling (SGNS) (Mikolov et al., 2013) and fastText (Bojanowski et al., 2017). For PPMI-SVD and Skip-gram embeddings, we use hyperwords (Levy et al., 2015). We run the tools with standard settings on the additional texts and use the resulting word representations as an additional feature for Marmot (Müller and Schütze, 2015). While fastText allows to obtain representations for OOV words by summing the representations of character n-grams, we do not use this feature as Marmot needs to be trained with a fixed set of word representations. However, learning representations for words and character n-grams simultaneously is — according to Bojanowski et al. (2017)—beneficial for small datasets and improves the representations for rare words. It might help in the case of spelling variation as well.

Table 12 shows the results. Using the *hist*-feature set leads to the best results. All three em-

	ReM					ReN			
Tagger	strict	norm	simple	spellvar	spellvar$_{pos}$	strict	simple	spellvar	spellvar$_{pos}$
Marmot-hist	*85.86*	87.02*	**87.15***	86.47*	85.99	83.71	84.70	84.38	84.01
Marmot-orig	84.28	86.87*	85.88	85.14#	84.91#	*84.15*	84.45	**84.81***	84.65*

Table 11: Spelling variant reduction (test). '*' marks a significant improvement over tagging the strict version with Marmot-*hist* for ReM and Marmot-*orig* for ReN, '#' marks a significant loss in accuracy.

bedding approaches lead to improvements that are comparable with the results obtained by automatic normalization and spelling variant detection. fastText gives the best results for both datasets, but only for ReM the improvement is significant over tagging the strict version with Marmot-*hist*. The reason for the small improvement might be the rather small amount of unlabeled data. Tuning the hyperparameters of the embedding methods, e.g. reducing the dimensionality, might yield further improvements.

We also combine the embedding feature with spelling variant detection trained with lemma, POS and morphology, see Table 13. By combining both, again there is a small improvement. For ReM, this improvement is significant for all types of embeddings. For ReN, there is no significant improvement over Marmot-*hist*, indicating that the combination of infix features and word representations also covers a lot of the spelling variation.

For Middle High German, there exists an independently created POS model for the TreeTagger (Echelmeyer et al., 2017). The tagset used to train the model is coarse-grained, consisting of only 18 tags. We use the tags predicted by this tagger as an additional feature. We expect this tagger to work better on the normalized version of ReM than on the other versions, because the TreeTagger model has been trained on data from the *Mittelhochdeutsche Begriffsdatenbank* (Middle High German Conceptual Database)[8], which contains texts from editions that consist of normalized Middle High German. This is confirmed by the results of training Marmot-*orig* and Marmot-*hist* on the strict version of the texts adding as additional feature the tag produced by the TreeTagger model a) on the strict version, or b) on the normalized version, see Table 14.

When using the TreeTagger tags generated on the strict version, there is no significant improvement compared to tagging the text without the additional feature. We conclude that we need normalized input to get good improvement from the tagger in this particular setting. Hence we also train our tagger on the normalized data. As Marmot-*orig* performs better for the normalized data, we only use the original features. Table 15 shows the results when training the tagger on normalized data with added tags as predicted by the TreeTagger model. Adding the tags as features leads to significant improvements only for automatic normalization.

8 Conclusion and further work

In this paper, we have investigated training a part-of-speech (POS) tagger for historical German in a low-resource setting—that is training with only about 12,000 tokens—and looked into different ways to deal with spelling variation.

Normalization as a means to reduce spelling variation has the biggest potential to improve the tagging accuracy: Using gold normalization, tagging accuracy improves from 84.05% to 89.71% on the development set for ReM. It also enables the usage of tools for normalized Middle High German, exemplified with an existing tagger, increasing the tagging accuracy to 90.24%. However, when using automatic normalization with a character-based SMT model trained on about 12,000 tokens, the tagging accuracy drops to 87.35% with the additional tagger for Middle High German and to 85.39% without. While this is an increase of slightly over 1% in tagging accuracy, we evaluated alternative ways to deal with spelling variation that result in similar improvements without the requirement of training a normalizer.

Firstly, we looked into using a specialized feature set for the POS tagger. By adding all character n-grams instead of only prefixes and suffixes as features for rare words and adapting the rare word threshold, we were able to improve tagging accuracy to 85.06% for ReM. For ReN, the dataset with less variation, the improvment in accuracy is not significant on the development set. On the test set, the original feature set even leads to better results. This shows that when training a POS tagger on data

[8] http://mhdbdb.sbg.ac.at/

	ReM			ReN		
Tagger	PPMI-SVD	SGNS	fastText	PPMI-SVD	SGNS	fastText
Marmot-hist	85.26	85.22	**85.95***	86.21	86.32	**86.37**
Marmot-orig	84.30#	84.25#	85.24	85.52#	85.56#	85.82

Table 12: Tagging with word embedding feature (development).'*' marks a significant improvement over tagging the strict version with Marmot-*hist*, '#' marks a significant loss in accuracy.

	ReM			ReN		
Tagger	PPMI-SVD	SGNS	fastText	PPMI-SVD	SGNS	fastText
Marmot-hist	85.69*	85.67*	**86.20***	86.35	**86.55**	86.25
Marmot-orig	85.08	85.04	85.78*	85.89	85.97	85.97

Table 13: Tagging with word embedding feature and spelling variant detection (development).'*' marks a significant improvement over tagging the strict version with Marmot-*hist*.

	TreeTagger	
Tagger	strict	norm
Marmot-hist	85.21	88.16*
Marmot-orig	84.83	87.83*

Table 14: Tagging Middle High German (ReM) with additional POS tags (development).'*' marks a significant improvement over tagging the strict version with Marmot-*hist*.

Tagger	Normalization	ReM
Marmot-orig	automatic	87.35*
	gold	**90.24**

Table 15: Tagging normalized Middle High German (ReM) with additional POS tags (development).'*' marks a significant improvement over tagging without the additional POS feature.

with spelling variation, it pays off to use specialized features instead of simply using the available feature set developed for standardized languages. However, infix features—as added in this paper—only help for data with a certain amount of spelling variation.

Secondly, we evaluated alternatives for normalization to reduce spelling variation. By applying rule-based simplification in combination with specialized features, tagging accuracy was improved to 86.02% for ReM and to 86.49% for ReN. So, by creating a small set of rewrite rules to reduce variation, it is possible to improve tagging accuracy more than with automatic normalization. As another alternative to normalization, we evaluated the substitution of OOV words with automatically detected spelling variants. The system used for spelling variant detection only needs POS tags to extract a noisy training set of variant pairs. In combination with the specialized feature set, we reached an accuracy of 85.46% for ReM, which is similar to the accuracy reached with automatic normalization, and 86.39% for ReN. While simplification and substitution of spelling variants do not lead to the same amount of improvements in tagging accuracy as using gold normalization does, they can be performed automatically with less effort. Compared to the automatic normalizer trained on 12,000 tokens, we were able to reach the same accuracy without needing any other training data than the data for the POS tagger. Thus, even without any additional data or resources, this approach can be used to improve tagging accuracy.

An addition to these approaches is to use word representations extracted from unlabeled text. This especially applies to modern user-generated data, where unlabeled data is easily available in large quantities. For historical texts, unlabeled data is not as easily available. However, if it is available, it is straightforward to improve tagging accuracy by word embeddings trained on this background corpus. In combination with the specialized feature set and substitution of spelling variants, tagging accuracy improved to 86.20% for ReM and 86.55% for ReN.

While we concentrated on a low-resource setting in this paper, for further work, it will be interesting to see how these methods scale with more training data and more available resources.

Resources

The scripts used to run the experiments for this paper are available at `https://github.com/fab-bar/paper-KONVENS2018`.

Acknowledgments

The first author was supported by the German Research Foundation (DFG), grants SCHR 999/5-3 and DI 1558/5-3. We would like to thank the anonymous reviewers for their helpful remarks.

References

Yvonne Adesam and Gerlof Bouma. 2016. Old Swedish part-of-speech tagging between variation and external knowledge. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 32–42, Berlin, Germany.

Fabian Barteld, Ingrid Schröder, and Heike Zinsmeister. 2015. Unsupervised regularization of historical texts for POS tagging. In *Proceedings of the Workshop on Corpus-Based Research in the Humanities (CRH)*, pages 3–12, Warsaw, Poland.

Fabian Barteld, Sarah Ihden, Katharina Dreessen, and Ingrid Schröder. 2018. HiNTS: A tagset for Middle Low German. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Fabian Barteld. 2017. Detecting spelling variants in non-standard texts. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 11–22, Valencia, Spain.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Marcel Bollmann. 2013. POS tagging for historical texts with sparse training data. In *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse*, pages 11–18, Sofia, Bulgaria.

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, Seattle, Washington, USA.

Stefanie Dipper, Karin Donhauser, Thomas Klein, Sonja Linde, Stefan Müller, and Klaus-Peter Wegera. 2013. HiTS: ein Tagset für historische Sprachstufen des Deutschen. *JLCL*, 28(1):1–53.

Stefanie Dipper. 2010. POS-tagging of historical language data: First experiments. In *Proceedings of the 10th Conference on Natural Language Processing (KONVENS 2010)*, pages 117–121, Saarbrücken, Germany.

Nora Echelmeyer, Nils Reiter, and Sarah Schulz. 2017. Ein PoS-Tagger für "das" Mittelhochdeutsche. In *Dhd 2017. Digitale Nachhaltigkeit. Konferenzabstracts*, pages 141–147, Bern, Switzerland.

Allen L. Edwards. 1948. Note on the "correction for continuity" in testing the significance of the difference between correlated proportions. *Psychometrika*, 13(3):185–187.

Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 2012. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484.

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*, pages 209–212, Prague, Czech Republic.

Thomas Klein, Klaus-Peter Wegera, Stefanie Dipper, and Claudia Wich-Reif. 2016. Referenzkorpus Mittelhochdeutsch (1050–1350), Version 1.0, https://www.linguistics.ruhr-uni-bochum.de/rem/. ISLRN 332-536-136-099-5.

Prasanth Kolachina, Martin Riedl, and Chris Biemann. 2017. Replacing OOV words for dependency parsing with distributional semantics. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 11–19, Gothenburg, Sweden.

Mariya Koleva, Melissa Farasyn, Bart Desmet, Anne Breitbarth, and Véronique Hoste. 2017. An automatic part-of-speech tagger for Middle Low German. *International Journal of Corpus Linguistics*, 22(1):107–140.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Vladimir Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association of Computational Linguistics*, 3:211–225.

Nikola Ljubešić, Katja Zupan, Darja Fišer, and Tomaž Erjavec. 2016. Normalising Slovene data: historical texts vs. user-generated content. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, pages 146–155, Bochum, Germany.

Pavel Logačev, Katrin Goldschmidt, and Ulrike Demske. 2014. POS-tagging historical corpora: The case of Early New High German. In *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT-13)*, pages 103–112, Tübingen, Germany.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR) 2013, Workshop Track.*

Fantine Mordelet and Jean-Philippe Vert. 2014. A bagging SVM to learn from positive and unlabeled examples. *Pattern Recognition Letters*, 37:201–209.

Thomas Müller and Hinrich Schütze. 2015. Robust morphological tagging with word representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 526–536, Denver, Colorado, USA.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

ReN-Team. 2018. Referenzkorpus Mittelniederdeutsch/Niederrheinisch (1200–1650). Archived in Hamburger Zentrum für Sprachkorpora. Version 0.6. Publication date 2018-03-07. http://hdl.handle.net/11022/0000-0007-C64C-5.

Yves Scherrer and Nikola Ljubešić. 2016. Automatic normalisation of the Swiss German ArchiMob corpus using character-level machine translation. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, pages 248–255, Bochum, Germany.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Universities of Stuttgart und Tübingen, Germany.

Helmut Schmid and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 777–784, Manchester, UK.

Sarah Schulz. 2018. *The Taming of the Shrew. Non-Standard Text Processing in the Digital Humanities*. Ph.D. thesis, University of Stuttgart, Germany.

Erik Tjong Kim Sang, Marcel Bollmann, Remko Boschker, Francisco Casacuberta, Feike Dietz, Stefanie Dipper, Miguel Domingo, Robe van der Goot, Marjo van Koppen, and Nikola Ljubešić. 2017. The CLIN27 shared task: Translating historical text to contemporary language for improving automatic linguistic annotation. *Computational Linguistics in the Netherlands*, 7:53–64.

Rob van der Goot, Barbara Plank, and Malvina Nissim. 2017. To normalize, or not to normalize: The impact of normalization on part-of-speech tagging. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 31–39, Copenhagen, Denmark.

Yi Yang and Jacob Eisenstein. 2016. Part-of-speech tagging for historical English. In *Proceedings of NAACL-HLT 2016*, pages 1318–1328, San Diego, California, USA.