Improving historical spelling normalization with bi-directional LSTMs and multi-task learning

Marcel Bollmann
Department of Linguistics
Ruhr-Universität Bochum
Germany
bollmann@linguistics.rub.de

Anders Søgaard
Dpt. of Computer Science
University of Copenhagen
Denmark
soegaard@hum.ku.dk

Abstract

Natural-language processing of historical documents is complicated by the abundance of variant spellings and lack of annotated data. A common approach is to normalize the spelling of historical words to modern forms. We explore the suitability of a deep neural network architecture for this task, particularly a deep bi-LSTM network applied on a character level. Our model compares well to previously established normalization algorithms when evaluated on a diverse set of texts from Early New High German. We show that multi-task learning with additional normalization data can improve our model's performance further.

1 Introduction

Interest in computational processing of historical documents is on the rise, as evidenced by the growing field of digital humanities and the increasing number of digitally available resources of historical data. Spelling normalization, i.e. the mapping of historical spelling variants to standardized/modernized forms, is often employed as a pre-processing step to allow the utilization of existing tools for the respective modern target language (Piotrowski, 2012).

Training data for supervised learning of spelling normalization is typically scarce in the historical domain. Furthermore, dialectal influences and even individual preferences of an author can have a huge impact on the spelling characteristics in a particular text, meaning that even training data from other corpora of the same language and time period cannot always be reliably used.

Algorithms have often been developed with this fact in mind, e.g. by being based on some form of phonetic, graphematic, or semantic similarity measure (Jurish, 2010; Bollmann, 2012; Amoia and Martinez, 2013). On the other hand, neural networks – and particularly deep networks with several hidden layers – are assumed to work best when trained on large amounts of data. It is therefore not clear whether neural networks are a good choice for this particular domain.

We frame spelling normalization as a character-based sequence labeling task, and explore the suitability of a deep bi-directional long short-term memory model (bi-LSTM) in this setting. By basing our model on individual characters as input, along with performing some basic preprocessing (e.g., downcasing all characters), we keep the vocabulary size small, which in turn reduces the model's complexity and the amount of data required to train it effectively. We show that this model outperforms both the existing normalization tool Norma (Bollmann, 2012) and a CRF-based tagger when evaluated on a diverse dataset from Early New High German.

Furthermore, we experiment with a multi-task learning setup using auxiliary data that has similar, but not identical spelling characteristics to the target text. We show that using bi-LSTMs with this multi-task learning setup can improve normalization accuracy further, while Norma and CRF do not profit much from the additional data in a traditional setup.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 131–139, Osaka, Japan, December 11-17 2016.

2 Datasets

We use a total of 44 texts from the Anselm corpus (Dipper and Schultz-Balluff, 2013) of Early New High German.[1] The corpus is a collection of several manuscripts and prints of the same core text, a religious treatise. Although the texts are semi-parallel and share some vocabulary, they were written in different time periods (between the 14th and 16th century) as well as different dialectal regions, and show quite diverse spelling characteristics. For example, the modern German word *Frau* 'woman' can be spelled as *fraw/vraw* (Me), *frawe* (N2), *frauwe* (St), *fraüwe* (B2), *frow* (Stu), *vrowe* (Ka), *vorwe* (Sa), or *vrouwe* (B), among others.[2]

All texts in the Anselm corpus are manually annotated with gold-standard normalizations following guidelines described in Krasselt et al. (2015). For our experiments, we excluded texts from the corpus that are shorter than 4,000 tokens, as well as a few texts for which annotations were not yet available at the time of writing (mostly Low German and Dutch versions). Nonetheless, the remaining 44 texts are still quite short for machine-learning standards, ranging from about 4,200 to 13,200 tokens, with an average length of 7,353 tokens.

For all texts, we removed tokens that consisted solely of punctuation characters. We also lowercase all characters, since it helps keep the size of the vocabulary low, and uppercasing of words is usually not very consistent in historical texts.

2.1 Conversion to labeled character sequences

Normalization is annotated on a word level; to reframe the problem as a character-based sequence labeling task, we need to align the historical wordforms and their normalizations on a character level. Ideally, we would like these alignments to be linguistically plausible, i.e., characters that most likely correspond to each other (e.g., historical *j* and modern *i*, as in *jn – ihn* 'him') should be aligned whenever possible.

The Levenshtein algorithm (Levenshtein, 1966) can be used to produce alignments that preferably align identical characters, but is ambiguous when multiple alignments with the same Levenshtein distance exist. We therefore use iterated Levenshtein distance alignment (Wieling et al., 2009), which uses pointwise mutual information on aligned segments to estimate statistical dependence, and favors alignments of characters that tend to cooccur often within the dataset. Since different texts can use the same characters in different ways, we perform this iterated alignment separately for each text.

A difficulty of these alignments is that the two wordforms can be of different lengths. We introduce a special epsilon label (ϵ) whenever a historical character is not aligned to any character in the normalization. We cannot do that for the inverse case, since the historical characters are our units of annotation and therefore need to be fixed, so we choose to perform a leftward merging of normalized characters whenever they are not aligned to any character in the historical wordform. For the word-initial case, we introduce a special "start of word" symbol (_). This symbol is prepended to each word during both training and testing, and is assigned the epsilon label during training when there is no word-initial insertion.

Here is an example of the final character sequence representation for the word pair *vsfuret – ausführt* '(he) leads out':

(1) _ v s f u r e t
a u s f üh r ϵ t

A consequence of this approach is that our labels cannot only be characters, but also combinations of characters (such as *üh* in the example above); our label set is therefore potentially unbounded. However, we found that this is not much of a problem in practice, since these cases tend to be comparatively rare.

3 Model

Our model architecture consists of: (i) an embedding layer for the input characters; (ii) a stack of bi-directional long short-term memory units (bi-LSTMs); and (iii) a final dense layer with a softmax activation to

[1] https://www.linguistics.rub.de/anselm/

[2] Abbreviations in brackets refer to individual texts using the same internal IDs that are found in the Anselm corpus.

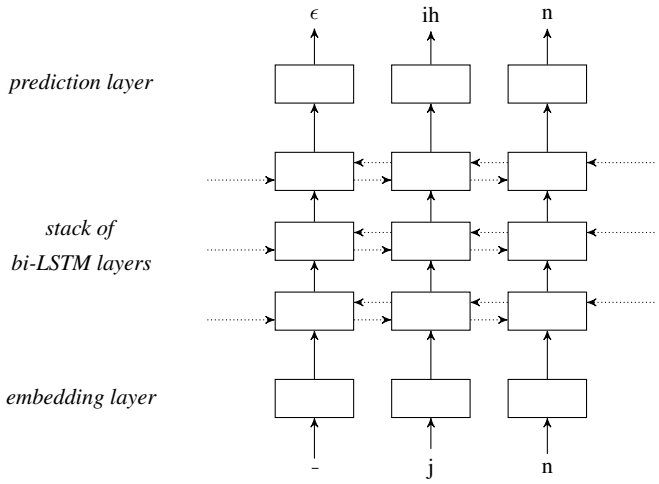

Figure 1: Flow diagram of the bi-LSTM character sequence labeling model, unrolled for time, for the word pair *jn – ihn* 'him'.

generate a probability distribution over the output classes at each timestep. An illustration of the model can be found in Figure 1.

The embedding layer maps one-hot input vectors (representing historical characters) to dense vectors. We did not use pre-trained embeddings; the embeddings are initialized randomly and learned as part of the regular network training process.

LSTMs (Hochreiter and Schmidhuber, 1997) are a form of recurrent neural network (RNN) designed to better learn long-term dependencies, and have proven advantageous to plain RNNs on many tasks. Bi-directional LSTMs read their input in both normal and reversed order, allowing the model to learn from both left and right context at each input timestep. A stack of bi-LSTMs, or a deep bi-LSTM, is a configuration of several bi-LSTM units so that the output of the ith unit is the input of the $(i+1)$th unit. In our model, we use a stack of three bi-LSTM layers.

The final dense layer is used to generate the output predictions, based on a linear transformation of the bi-LSTM outputs for each timestep followed by a softmax activation. We train the model by minimizing the cross-entropy loss across all output characters, and using backpropagation to update the weights in all layers (including the embedding layer). During prediction, we generate output labels in a greedy fashion, choosing the label with the highest probability for each timestep.

3.1 Multi-task learning setup

In multi-task learning (MTL), the performance of a model on a given task is improved by additionally training it on one or more auxiliary tasks (Caruana, 1993). For our bi-LSTM model, this means that all layers of the model are shared between the tasks apart from the final prediction layer, which is kept separate for the main and auxiliary tasks. This way, errors in an auxiliary task that are backpropagated through the network also affect the prediction of the main task, helping to regularize the network's weights and prevent overfitting.

Multi-task learning with (deep) neural network architectures was shown to be effective for a variety of NLP tasks, such as part-of-speech tagging, chunking, named entity recognition (Collobert et al., 2011); sentence compression (Klerke et al., 2016); or machine translation (Luong et al., 2016).

In our experiments, we regard spelling normalization within a target domain (i.e., a given historical text) as our main task, while using normalization within related domains (i.e., texts from a similar time period, but with distinct spelling characteristics) as our auxiliary task. During training, we alternate between training on a random instance from the main and the auxiliary tasks.

3.2 Hyperparameters

We set aside one of the texts (B) from the Anselm corpus for testing different hyperparameter configurations. On this text, we achieved the best results with a dimensionality of 128 for the embedding and bi-LSTM layers, using a dropout of 0.1, and training the model for 30 iterations. These settings were subsequently used for all further experiments.

3.3 Other models used for comparison

For comparison, we also train and evaluate with the Norma tool described by Bollmann (2012), since it was originally developed for the Anselm corpus and the implementation is publicly available.[3] However, Norma actually consists of a combination of three different normalization methods, one of which is a simple wordlist mapping of historical tokens to normalized forms. Since this wordlist mapping is conceptually very simple and could easily be added to our (or any other) normalization method, we exclude it for the comparison, and only use Norma's remaining two algorithms (which we denote Norma*).

Additionally, since we frame the problem as a sequence labeling task, we compare our results to a simple sequence labeling model using conditional random fields (CRF). The CRF model gets the same input/output sequences as our bi-LSTM model (cf. Sec. 2.1), and uses the two preceding and following characters from the historical wordform as additional features. Implementation was done with CRFsuite (Okazaki, 2007) using the averaged perceptron algorithm for training.

4 Evaluation

We evaluate our model separately for each text in our dataset. From each text, we use 1,000 tokens as our evaluation set, set aside another 1,000 tokens as a development set (which was not currently used), and train on the remaining tokens (between 2,000 and 11,000, depending on the text). Both CRF and our bi-LSTM model get their input as character sequences (as described in Sec. 2.1), while Norma requires full words as input.

For the multi-task learning setup, we randomly sample from all Anselm texts and regard each text as its own task. Effectively, we are learning a joint model over all Anselm texts with shared parameters but distinct prediction layers, while viewing the text we are currently evaluating on as our main task and the others as auxiliary tasks. The MTL setup is only applicable to our bi-LSTM model; however, since the auxiliary task consists of spelling normalization with data from the same corpus (although with a higher variety of different spelling characteristics compared to the target text), it is possible that the other methods could also profit from this additional training data. We therefore also evaluate Norma and CRF when the training sets have been augmented by 10,000 randomly sampled training examples from all texts.

4.1 Word accuracy

Evaluation results in terms of word-level accuracy are presented in Table 1.

Columns "S" show results for the traditional setup without multi-task learning. The basic bi-LSTM model performs better than Norma on 34 of the 44 texts. On average, there is an increase of 2.1 percentage points (pp), although the differences on individual texts vary wildly, from -2.9 pp (M5) to $+9.6$ pp (M), giving a standard deviation of 2.7 pp. The CRF model, on the other hand, is almost always worse than Norma, averaging a difference of -2.1 pp (± 2.0). This indicates that the reformulation of the task as character-based sequence labeling cannot alone be responsible for the bi-LSTM results, but the choice of a neural network architecture is crucial, too.

Columns "S+A" present the results when using the augmented training set. For bi-LSTM, this is the multi-task learning setup—using MTL improves the results by $+0.7$ pp (± 2.8) on average, but again there is a high variance within the individual scores. However, for the other methods, adding the 10,000 randomly selected samples to the training set actually decreases the average accuracy, by -0.4 pp for Norma and -2.0 pp for CRF. This is likely due to the fact that this additional training set introduces a variety of spelling characteristics that are not found in the target text. While Norma and

[3] https://github.com/comphist/norma

ID	Region	Tokens	Norma*		CRF		Bi-LSTM	
			S	S+A	S	S+A	S	S+A†
B	East Central	4,718	80.30%	77.80%	76.30%	72.80%	79.20%	**81.70%**
D3	East Central	5,704	80.50%	80.20%	77.20%	73.00%	80.10%	**81.20%**
H	East Central	8,427	82.70%	82.90%	78.60%	76.00%	**85.00%**	82.30%
B2	West Central	9,145	76.10%	77.60%	74.60%	71.70%	**82.00%**	79.60%
KÄ1492	West Central	7,332	77.50%	74.40%	74.80%	68.40%	**81.60%**	80.50%
KJ1499	West Central	7,330	77.00%	72.90%	73.50%	68.40%	**84.50%**	79.20%
N1500	West Central	7,272	76.70%	75.30%	72.70%	67.20%	79.00%	**79.20%**
N1509	West Central	7,418	78.10%	73.30%	74.30%	68.80%	**80.80%**	80.10%
N1514	West Central	7,412	78.30%	73.80%	72.20%	69.90%	79.00%	**80.10%**
St	West Central	7,407	72.60%	73.80%	70.30%	68.70%	**75.50%**	75.20%
D4	Upper/Central	5,806	75.60%	75.60%	72.40%	70.90%	76.50%	**76.60%**
N4	Upper	8,593	78.20%	78.10%	80.00%	78.40%	81.80%	**83.40%**
s1496/97	Upper	5,840	81.70%	83.40%	77.70%	76.90%	83.00%	**84.10%**
B3	East Upper	6,222	80.80%	80.60%	79.50%	79.10%	81.50%	**83.20%**
Hk	East Upper	8,690	77.80%	79.30%	78.20%	77.90%	80.90%	**82.20%**
M	East Upper	8,700	74.30%	74.40%	72.80%	68.40%	**83.90%**	80.90%
M2	East Upper	8,729	75.80%	76.00%	75.10%	72.40%	76.70%	**80.20%**
M3	East Upper	7,929	79.00%	79.70%	77.30%	74.10%	**80.40%**	79.60%
M5	East Upper	4,705	80.60%	80.70%	76.40%	78.30%	77.70%	**82.90%**
M6	East Upper	4,632	75.90%	76.30%	73.70%	74.40%	75.20%	**79.30%**
M9	East Upper	4,739	82.20%	81.50%	79.00%	76.90%	80.40%	**83.60%**
M10	East Upper	4,379	77.00%	78.60%	76.00%	75.80%	75.10%	**81.30%**
Me	East Upper	4,560	79.70%	80.10%	76.90%	75.50%	80.30%	**83.70%**
Sb	East Upper	7,218	78.00%	76.60%	75.70%	74.80%	**80.00%**	78.50%
T	East Upper	8,678	76.70%	78.60%	73.40%	72.20%	75.80%	**79.00%**
W	East Upper	8,217	75.90%	78.30%	78.20%	77.00%	**81.40%**	80.80%
We	East Upper	6,661	**83.10%**	81.50%	78.60%	75.80%	81.50%	**83.10%**
Ba	North Upper	5,934	79.80%	81.20%	80.20%	78.70%	80.70%	**82.80%**
Ba2	North Upper	5,953	81.40%	80.00%	78.10%	77.90%	82.50%	**84.10%**
M4	North Upper	8,574	76.90%	76.70%	75.70%	75.00%	79.40%	**82.30%**
M7	North Upper	4,638	79.40%	79.80%	75.60%	74.20%	78.20%	**82.10%**
M8	North Upper	8,275	78.50%	77.00%	78.20%	78.40%	81.10%	**82.50%**
n	North Upper	9,191	79.60%	81.30%	81.90%	78.20%	84.40%	**84.70%**
N	North Upper	13,285	75.50%	76.30%	71.70%	68.90%	**79.00%**	76.90%
N2	North Upper	7,058	82.20%	81.90%	80.30%	81.60%	**84.30%**	83.40%
N3	North Upper	4,192	79.10%	80.80%	76.40%	77.50%	77.60%	**84.20%**
Be	West Upper	8,203	75.50%	76.40%	75.30%	73.40%	**78.80%**	78.00%
Ka	West Upper	12,641	73.80%	74.10%	75.40%	72.80%	80.10%	**80.30%**
SG	West Upper	7,838	80.10%	79.90%	78.00%	76.80%	**81.70%**	80.90%
Sa	West Upper	8,668	72.60%	73.50%	71.90%	71.40%	76.10%	**76.50%**
St2	West Upper	8,834	73.20%	73.40%	73.20%	73.00%	78.20%	**79.90%**
Stu	West Upper	8,686	77.70%	77.10%	76.50%	72.10%	**79.40%**	77.00%
Sa2	West Upper	8,011	77.50%	77.90%	73.50%	73.30%	79.50%	**79.70%**
Le	Dutch	7,087	69.50%	60.30%	65.00%	55.80%	**75.60%**	67.50%
Average		7,353	77.83%	77.48%	75.73%	73.70%	79.90%	**80.55%**

Table 1: Word accuracy on the Anselm dataset, evaluated on the first 1,000 tokens; S = training set from the same text, S+A = like S, but augmented with 10,000 tokens randomly sampled from the other texts; † = Bi-LSTM (S+A) is the multi-task learning setup. Best results shown in bold.

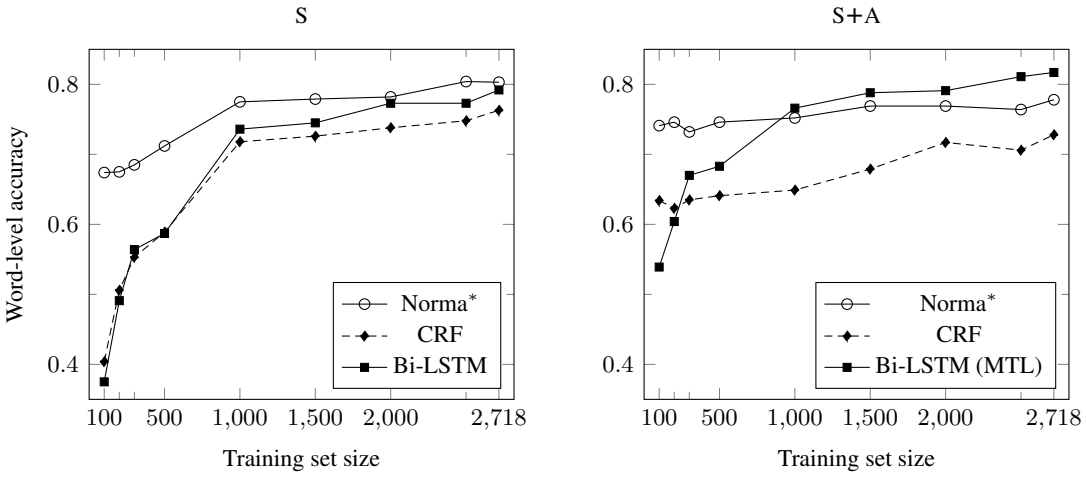

Figure 2: Word accuracy on the 'B' text for different sizes of the training set; left = train only on the training set from 'B' (S); right = use augmented training set/multi-task learning (S+A).

CRF cannot handle this out-of-domain training data well, the MTL setup can actually profit from it in many cases.

Table 1 also shows a rough classification of the dialectal regions from which the texts originate. There is a slight trend for multi-task learning to be advantageous on texts from the East and North Upper German regions, while for the Central and West Upper German texts, there are more instances of the standard bi-LSTM model (S) being better than the MTL model (S+A). This could either be due to linguistic properties of these dialectal regions, or due to the fact that East/North Upper German texts make up the majority of the dataset, thereby also featuring more prominently in the "S+A" settings.

The latter hypothesis is supported by the case of the 'Le' text, which is the only Dutch text in the sample (but which was nonetheless normalized to modern German in the corpus). Here, the "S+A" settings of the experiments all show a dramatic decrease in accuracy (up to -9.2 pp), suggesting that it is disadvantageous to augment the training set with samples that are too different from the target domain, even for the MTL setup.

In general, however, one of the bi-LSTM models is always best; there is only one text (We) for which Norma achieves an equal accuracy. This indicates that deep neural networks can be applied successfully to the spelling normalization task even with a comparatively small amount of training data. Also, we note that Norma always requires a lexical resource which it uses to filter results, while we do not.

4.2 Effect of training set size

In our evaluation, we use all but the first 2,000 tokens from a text for training (cf. the beginning of Sec. 4). Consequently, the training sets for each text are of different sizes. We calculate Spearman's rank correlation coefficient (ρ) between the size of the training sets and the normalization accuracy for each column in Table 1. We find no significant correlation for the CRF and bi-LSTM models ($|\rho| < 0.25$), although there seems to be a moderate *inverse* correlation for the Norma results ($\rho \approx -0.48$ on Norma "S"). The reasons for this are beyond the scope of this paper, though.

The question of how much training data is needed to effectively train a model is particularly relevant for historical spelling normalization, since training data can be very sparse in this domain. We therefore choose to evaluate each method in a scenario where we consider a single text, but vary the size of the training set, to estimate how well they perform with fewer data.

Figure 2 shows the results for different training set sizes on the 'B' text. Not surprisingly, when training on only 100 tokens, accuracy is bad ($< 41\%$) for CRF and bi-LSTM. Norma, on the other hand, already achieves 67.4% in this scenario. The biggest gains for all three methods can be seen for training set sizes between 100 and 1,000 tokens—for larger set sizes, the gains become less, and all three methods

are within close range of each other.

For the "S+A" scenario, all models have noticeably higher accuracy even with only 100 tokens from the 'B' text. However, the increases for Norma and CRF are not as high as in the "S" scenario; this is not surprising, since the total training set for these methods always contains at least 10,000 tokens (from the auxiliary set), and it is only the proportion of tokens coming from the 'B' text that increases. The bi-LSTM model with multi-task learning behaves differently, though: while it starts off as the weakest model (on 100 tokens), it is the best model when training on 1,000 tokens or more.

These learning curves illustrate that the MTL setup is fundamentally different from adding the auxiliary data to the training set normally, as is the case with CRF and the Norma tool. They also show that our bi-LSTM models can be better than or at least competitive with CRF/Norma for training set sizes as low as 1,000 tokens.

4.3 Multi-task learning with grapheme-to-phoneme mappings

It is conceivable to use different tasks than historical spelling normalization as the auxiliary task in a multi-task learning setup. In particular, we also experimented with grapheme-to-phoneme mapping as the auxiliary task, since it can be seen as a similar form of character-based sequence transduction.

For our dataset, we used the German part of the CELEX lexical database (Baayen et al., 1995), particularly the database of phonetic transcriptions of German wordforms. The database contains a total of 365,530 wordforms with transcriptions in DISC format, which assigns one character to each distinct phonological segment (including affricates and diphthongs). For example, the word *Jungfrau* 'virgin' is represented as `'jUN-frB`. We randomly sampled 4,000 tokens from this dataset for our experiment, and used the same algorithm as for the historical data to convert these mappings to a character-based sequence representation (cf. Sec. 2.1).

The evaluation, however, showed no real benefit of this MTL setup compared to the bi-LSTM model without MTL. While accuracy increased for some texts by up to 2.6 pp, it decreased slightly for the majority of texts, averaging to a -0.4 pp difference to the basic model.

5 Related Work

Various methods have been proposed to perform spelling normalization on historical texts; for an overview, see Piotrowski (2012). Many approaches use edit distance calculations or some form of character-level rewrite rules, but require either hand-crafting of the rules (Baron and Rayson, 2008) or a lexical resource to filter their output (Bollmann, 2012; Porta et al., 2013).

A newer approach is the application of character-based statistical machine translation (Pettersson et al., 2013; Sánchez-Martínez et al., 2013; Scherrer and Erjavec, 2013). In contrast to our sequence labeling approach, these methods do not require a fixed character alignment between wordforms, but it is not clear whether this is actually an advantage. To our knowledge, a comparative evaluation between these methods and other approaches has not yet been done.

Azawi et al. (2013) present the only other approach we are aware of that applies neural networks to normalization of historical data. They also use bi-directional LSTMs, but differ from our approach in the way they perform alignment between historical and modern wordforms. More importantly, they evaluate their model on a single dataset, the Luther bible, which has much more regular spelling than the texts in the Anselm corpus and is also significantly longer: they use about 200,000 tokens for their training set.

6 Conclusion and Future Work

We presented an approach to historical spelling normalization using bi-directional long short-term memory networks and showed that it outperforms a CRF baseline and the Norma tool by Bollmann (2012) for almost all of the texts in our dataset, a diverse corpus of Early New High German, despite using a relatively low amount of training data (about 2,000 to 11,000 tokens) and not making use of a lexical resource (like Norma does). We showed further that multi-task learning with additional normalization data can improve accuracy with bi-LSTMs, while adding the same data to the training set of Norma and CRF does not help on average, and can even be detrimental.

Many improvements to this approach are conceivable. Character-based statistical machine translation has been successfully applied to spelling normalization (cf. Sec. 5), but we are not aware of any experiments with neural machine translation (Cho et al., 2014) on this domain. Using an encoder–decoder architecture, e.g. similar to Sutskever et al. (2014), would remove the need for an explicit character alignment (cf. Sec. 2.1) but could also make the model more complex and potentially more difficult to train, so it is unclear whether this would be an improvement to our approach.

With regard to multi-task learning, our results seem to indicate that for the auxiliary task, it is preferable to use data with similar characteristics to the data in the main task. On the other hand, depending on the language variety to be annotated, such data might not always be readily available. We would therefore like to do further experiments with auxiliary data from different corpora or even different string transduction tasks, to see if and under which conditions they can have a beneficial effect on the spelling normalization task.

Acknowledgments

Marcel Bollmann was supported by Deutsche Forschungsgemeinschaft (DFG), Grant DI 1558/4.

References

Marilisa Amoia and Jose Manuel Martinez. 2013. Using comparable collections of historical texts for building a diachronic dictionary for spelling normalization. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 84–89, Sofia, Bulgaria.

Mayce Al Azawi, Muhammad Zeshan Afzal, and Thomas M. Breuel. 2013. Normalizing historical orthography for OCR historical documents using LSTM. In *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*, pages 80–85. ACM.

R. Harald Baayen, Richard Piepenbrock, and Léon Gulikers. 1995. The CELEX lexical database (Release 2) (CD-ROM). Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

Alistair Baron and Paul Rayson. 2008. VARD 2: A tool for dealing with spelling variation in historical corpora. In *Proceedings of the Postgraduate Conference in Corpus Linguistics*.

Marcel Bollmann. 2012. (Semi-)automatic normalization of historical texts using distance measures and the Norma tool. In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, Lisbon, Portugal.

Rich Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the 10th International Conference on Machine Learning (ICML)*, pages 41–48.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, pages 103–111, Doha, Qatar.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Stefanie Dipper and Simone Schultz-Balluff. 2013. The Anselm corpus: Methods and perspectives of a parallel aligned corpus. In *Proceedings of the NODALIDA Workshop on Computational Historical Linguistics*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Bryan Jurish. 2010. More than words: using token context to improve canonicalization of historical German. *Journal for Language Technology and Computational Linguistics*, 25(1):23–39.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *Proceedings of NAACL-HLT 2016*, pages 1528–1533, San Diego, CA.

Julia Krasselt, Marcel Bollmann, Stefanie Dipper, and Florian Petran. 2015. Guidelines for normalizing historical German texts. *Bochumer Linguistische Arbeitsberichte*, 15.

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

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. arXiv:1511.06114v4.

Naoaki Okazaki. 2007. CRFsuite: a fast implementation of conditional random fields (CRFs). http://www.chokkan.org/software/crfsuite/.

Eva Pettersson, Beáta Megyesi, and Jörg Tiedemann. 2013. An SMT approach to automatic annotation of historical text. In *Proceedings of the NODALIDA Workshop on Computational Historical Linguistics*, Oslo, Norway.

Michael Piotrowski. 2012. *Natural Language Processing for Historical Texts*. Number 17 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool, San Rafael, CA.

Jordi Porta, José-Luis Sancho, and Javier Gómez. 2013. Edit transducers for spelling variation in Old Spanish. In *Proceedings of the NODALIDA Workshop on Computational Historical Linguistics*, Oslo, Norway.

Yves Scherrer and Tomaž Erjavec. 2013. Modernizing historical Slovene words with character-based SMT. In *Proceedings of the 4th Biennial Workshop on Balto-Slavic Natural Language Processing*, Sofia, Bulgaria.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS 2014)*, number 27, pages 3104–3112.

Felipe Sánchez-Martínez, Isabel Martínez-Sempere, Xavier Ivars-Ribes, and Rafael C. Carrasco. 2013. An open diachronic corpus of historical Spanish: annotation criteria and automatic modernisation of spelling. arXiv:1306.3692v1, 06.

Martijn Wieling, Jelena Prokić, and John Nerbonne. 2009. Evaluating the pairwise string alignment of pronunciations. In *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education (LaTeCH – SHELT&R 2009)*, pages 26–34, Athens, Greece.