Orthographic Normalization of Historical Texts

Pryce Bevan[1], Luke Gessler[2], and Michael Kranzlein[1]

[1]Georgetown University, Department of Computer Science
[2]Georgetown University, Department of Linguistics
{pwb8, lg876, mmk119}@georgetown.edu

Abstract

We explore several model architectures for historical text normalization based on word types. At the word type level, we observe that a baseline imposes an upper bound on model performance on word types seen in training data. Accordingly, we implement an LSTM model and a transformer model to predict unseen word types and incorporate the baseline's predictions for seen word types. We test these hybrid models on historical Spanish texts and achieve results that are on par with state-of-the-art models. We also note that the LSTM model outperforms the transformer model, though this is likely due to unresolved issues with our transformer implementation.

1 Introduction

In many cases, researchers require or prefer to work with standardized data. In this work, we examine historical text normalization (also called canonicalization)—"the problem of translating historical documents written in the absence of modern spelling conventions and making them amenable to search by today's scholars, processable by natural language processing models, and readable to laypeople" (Bollmann et al., 2018). Bollmann et al. explicitly mention spelling conventions, but this is a standardization problem that extends beyond modernizing orthography. More comprehensive approaches to historical text normalization also address syntactic and semantic aspects of language, as word order changes, phrases fall out of style, new punctuation conventions are adopted, and word meanings change over time, even if the orthography doesn't. These facets of historical text normalization, however, are more difficult problems to address. We address only orthographic normalization in this work.

1.1 Motivating the Normalization Problem

We experiment with Spanish data because it is readily available, but we propose Coptic as a strong motivating example of the importance of historical text normalization. In work done on Coptic, Schroeder and Zeldes discuss the features of a normalized text, including "word segmentation and sentence segmentation based on modern editorial standards; standardized spelling of words throughout the text...; punctuation based on modern standards; removal of ancient strokes, punctuation, and/or accents; standardization of abbreviations" (Schroeder and Zeldes, 2013). This type of normalization performs a critical role in a manuscript-to-analysis pipeline (Zeldes and Schroeder, 2016). In short, there are many analyses digital humanists wish to pursue that orthographic irregularities are an obstacle to.

1.2 Our Research Goals

At the outset of this project, we aimed to achieve results on par with current state-of-the-art methods for orthographic historical text normalization. While we cannot conduct an apples-to-apples comparison with recent work (since we use a different dataset in a different language), we find generally similar results using a bidirectional LSTM recurrent neural network on Spanish data from the past few centuries (Graves and Schmidhuber, 2005). We attempted a transformer-based approach, but were not able to succeed in getting this model to achieve strong performance (Vaswani et al., 2017). This may be due to a flawed implementation or the use of a dataset that is too small.

2 Related Work

Historical text normalization has received renewed attention as neural approaches continue to gain traction in the NLP community. Over the past ten years, the state of the art has seen an evolution from Hidden Markov Models and rule-based models to simple neural networks to more context-aware recurrent neural networks. Most recently, there is a shift toward attention and transformers.

2.1 HMM, Levenshtein Edit Distance, and Rule-Based Methods

In 2010, Jurish described a Hidden Markov Model approach to incorporate context, that is, to allow a token's normalization to be decided on the basis of not just the one token, but the total context of the token in question as well as its neighbors to the left and right. This model achieved high precision and recall.

Jurish, in addition to Pettersson et al., have also both experimented with Levenshtein edit distance as a useful factor for determining the best candidate normalization. Informally, Levenshtein edit distance is the number of changes required to make one string equal to another (Levenshtein, 1966).

Rule-based methods have been around for a long time and remain an efficient approach for decent performance. (Bollmann et al., 2011; Pettersson, 2016; Zeldes and Schroeder, 2016; Schneider et al., 2017). but more recently, state-of-the-art performance tends to come from neural approaches.

2.2 Attention and Multi-Task Learning

In 2017, Bollmann et al. explored multi-task learning in the context of historical text normalization. They found that the utility of the attention could be achieved by learning the auxiliary task of pronouncing, that is, they constructed a model that jointly learned the orthography and a grapheme-to-phoneme mapping. In 2019, Bollmann et al. expanded upon this work by adding additional auxiliary tasks and A similar group of authors expanded on this work.

3 Implementation

Our code is available on GitHub at https://github.com/lgessler/notnaughtknotnauts-normalize.

3.1 Dataset

Many widely-used corpora in the literature were unfortunately not available. In some cases, it wasn't even clear exactly which corpus had been used (e.g. in Pettersson (2016)). In others, the corpus was not easily obtainable. The Corpus of Early English Correspondence[1], for example, is only available on CD-ROM after submission of a payment via post to Europe. In the end, we chose to use the only suitable and freely-available corpus we could find, the Post Scriptum corpus (Vaamonde et al., 2014).

The Post Scriptum corpus contains epistolary texts in Spanish and Portuguese, ranging from the 16th century to the 19th century in origin. These documents are unpublished and represent a diverse set of social backgrounds. For our experiments, we use only the Spanish texts, totaling 2368 documents with around 400,000 tokens. Each original document is included in its original orthographic form, and is also provided in modern orthography, which was manually produced by a human annotator. Our models ingest an original token as input and predict the best modern equivalent.

> *por me hazer md me ebye el bonete q conpre aqui*
>
> *Por me hacer merced, me envíe el bonete que compré aquí.*
>
> Figure 1: A sample of unnormalized text with its normalized version from the Post Scriptum corpus.

Some preprocessing was necessary in order to get the document-pairs into the form that we wanted them in: a list of tokens, both n tokens long, where each token at index i in either document is "the same" token. First, all metadata was removed from the documents, leaving only the body of the text. Next, text was lowercase normalized, and all punctuation was removed.[2] Then, the text was whitespace-tokenized.

At this point, we might sometimes be left with neatly even lists of tokens, but unfortunately, the same tokenization was not always reproduced exactly during the normalization process: apparently, some tokens in the original document were in a many-to-one relationship with tokens in the normalized document ($t_{\text{orig},i} \rightarrow t_{\text{norm},i}, \ldots, t_{\text{norm},i+k}$), or vice versa ($t_{\text{orig},i}, \ldots, t_{\text{orig},i+k} \rightarrow t_{\text{norm},i}$). This could have arisen for a variety of reasons: mistranscriptions, the expansion of multi-word abbreviations, differences in spacing conventions, etc.

In an effort to mitigate this as much as possible, we work with only the documents whose original

[1] http://www.helsinki.fi/varieng/CoRD/corpora/CEEC/

[2] This primarily affected the modernized writings, as the original orthographies did not contain much punctuation.

and normalized versions are the same length. It's important to note that while this does not guarantee correct one-to-one alignments (one document might "get ahead" early on then get caught back up to towards the end), a manual observation of several thousand token pairs indicated generally good alignments.

Once the data was preprocessed, we used an 80/10/10 split for training, validation, and testing.

3.2 Baseline

The baseline we used to evaluate our systems against simply takes the most commonly observed normalized token for a given original token. I.e., at each index i in the document being normalized, we take:

$$\hat{t}_{\text{norm},i} = \arg\max_{x} \text{freq}(t_{\text{orig,i}}, x)$$

where the freq function returns the number of occurrences of every token-pair in the entire corpus. If freq is 0, the baseline system produces a special unknown token for unseen inputs.

Although simple, the baseline performs well, and can be thought of as an upper bound for the performance of any system performing token-by-token normalization, under the assumption that all inputs will have been seen during training: as long as the training corpus is sufficiently big, it is highly likely that for any given input token, the most common output token in the testing corpus will also be the most common output token in the training corpus. However, a statistical system is still worth pursuing, because the baseline is powerless if an input token has not been seen before, and it is likely that any unseen tokens will have structure a statistical system would likely be able to take advantage of.

3.3 LSTM Model

Our first statistical system was an LSTM-based model implemented in Keras. It consists of an embedding layer, a bidirectional LSTM layer, and a softmax layer, with dropout layers in between them. The sequence presented to the LSTM is that of the characters in the token, with special characters added for the start and end of the word, as well as any padding characters necessary to make the token fit in its batch.

3.4 Transformer Model

Our second statistical system was a transformer-based model. It was implemented using AllenNLP. The encoder was a `StackedSelfAttentionEncoder`, provided in the standard distribution of AllenNLP. The decoder was a modified version of AllenNLP's `SimpleSeq2Seq` predictor that was extended so it could calculate accuracy in addition to loss. This extension, which seems like it should have been quite simple, actually took a lot of time, as AllenNLP's abstractions make implementing another metric for a model a complicated matter, requiring code changes in multiple places. More detail will be presented in the discussion, but we were ultimately unable to get satisfactory results with the transformer model.

4 Results

We evaluated our performance using the standard metric in the literature for normalization: character-based accuracy. Only "real" characters were considered when calculating the score: special tokens for padding and marking the beginning and end of a token were ignored. Accuracy was measured in a held-out test corpus.

Following another convention in the literature, we tested both the baseline system and our statistical systems independently, and also tested a hybrid system that would use the baseline if the token had been seen before, and otherwise use the statistical model if the token had not been seen before. These were our best results after several dozen experiments, with the LSTM system hyperparameters being an embedding size of 300, and an LSTM hidden size of 100:

System	Accuracy
Baseline	85.42%
LSTM only	84.88%
LSTM + Baseline hybrid	90.24%
Transformer	17.42%

We considered whether training and testing only on individual centuries might have helped performance, the idea being that there might be different orthographic alternations in each century, and that these century-specific alternations might even be contradictory. However, when we trained and tested on individual centuries, we found that performance did not improve or suffer. It's difficult to know exactly why this was, but it's possible that there were only a negligible number of

century-specific alternations, or that there were a significant number of century-specific alternations but that degradations caused by smaller amount of training data for each century counterbalanced any improvements that might have been had.

We also attempted to give the LSTM model some additional context, hoping that it would be able to help in the cases where a single unnormalized token could map to potentially many normalized tokens. We added a parameter that modified the input sequence so that it would include not just the input token but also n tokens to the left and right of the token, separated by special characters to represent the token boundary. We found that this had no effect for $n = 1$, and that it caused severe degradations for $n > 1$.

The degradation is easily explainable: the expansion in input size increases the dimensionality of the model by quite a bit, and the reason why this kind of context isn't helping is probably that if there *is* a word in the sentence that could help disambiguate an input token, then it is quite likely that it won't be within such a short range of the token. For instance, suppose an unnormalized verb could correspond to potentially many normalized forms because it was written without diacritics historically and gained diacritics in modern orthography that indicate agreement with its subject based on number, gender, etc. Occasionally the subject would be within this n token distance, but it would often not be, and so context would not help.[3]

5 Discussion

Direct comparison of our results to any other system was not possible, as there was no record we could find of another system being used with the Post Scriptum corpus. However, even with only our scores, we can speculate on how well we have done.

The LSTM-only system comes very close to the baseline's performance, which as discussed above can be thought of as an upper bound on performance for seen tokens. The difference is only $\sim$0.5%. Given this, it wouldn't seem a huge leap to claim that the LSTM probably succeeded in capturing the majority of the orthographic alternations.

The failure of the transformer system is difficult to explain. Frankly, since none of us understands how the model works very well, it is not even clear whether it is possible at all for a transformer to succeed in a task like this. Assuming that it is possible, there are several other possible explanations. First, perhaps none of our hyperparameter choices were sensible. Second, perhaps we simply didn't have enough training data. (Recall that the corpus had only 400,000 tokens.) Third, perhaps there was an implementation error in either the system itself or how we implemented accuracy. Strangely, accuracy was seen to decrease as loss decreased for the transformer system. One possible reason why this might be is that accuracy was only calculated over "real" characters while loss was calculated for all characters, including padding characters. Even so, the same could be said for the LSTM model, where accuracy behaved as expected.

6 Future Work

The biggest remaining question is what kind of errors were to blame for the remaining 10% accuracy. Unfortunately we did not have time to conduct an extensive error analysis, but there are several obvious places to look.

First, our naïve alignment strategy (assume that in each document-pair the indices for every token are the same) is, as we know, sometimes failing to properly pair off corresponding tokens. It would be a good idea to investigate more sophisticated heuristics to ensure proper alignments. Techniques like sequence matching and applying Levenshtein distance, as discussed in several of the related works, could be used to compare token pairs and better guarantee their relatedness, at the cost of some additional preprocessing time. Not only would yield higher-quality data, but it would allow us to obtain crude, but probably helpful, sentence alignments: the original documents do not have reliable sentence delimiters, but if token-level alignment were reasonably good, we could use the periods from the modernized documents to recover sentence boundaries in the original text.

Having somewhat sentence boundaries would also allow us to structure this problem as a machine translation task. Applying a machine translation approach blindly would probably not succeed very well, since this framing would lose several useful constraints that we know should hold (e.g. that tokens should never be swapped in order during the normalization process), but it would al-

[3] And the situation would be worse for languages where subjects do not typically appear next to the verb phrase, e.g. Hindi, where the word order is SOV.

low us to get the context that our token-level sequence to sequence approach lacks. A first step in this direction would be to conduct an error analysis and verify that the anticipated one-to-many normalization case is actually a significant source of error.

In addition, in future work, we would further evaluate the use of transformers and other attention-based approaches, as it seems likely that transformers can actually do better and that we are just doing something wrong in our implementation at the moment.

Finally, it would be interesting to see how well our approach works on corpora in different languages, including Portuguese, which is included in the Post Scriptum corpus. Spanish is SVO, has some but not too much inflection, and has relatively strict word order. Our approach might not generalize well to highly-inflected or agglutinative languages like Plains Cree or Turkish, where out-of-vocabulary items would be much more common, and our system, unmodified, would have to begin to serve as a morphological parser and presumably suffer for it.

Acknowledgments

We thank the attendees of the Georgetown EMNLP final project poster session for their thoughtful feedback.

References

Marcel Bollmann, Joachim Bingel, and Anders Søgaard. 2017. Learning attention for historical text normalization by learning to pronounce. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 332–344, Vancouver, Canada. Association for Computational Linguistics.

Marcel Bollmann, Natalia Korchagina, and Anders Søgaard. 2019. Few-Shot and Zero-Shot Learning for Historical Text Normalization. page 10.

Marcel Bollmann, Florian Petran, and Stefanie Dipper. 2011. Rule-Based Normalization of Historical Texts. In *Proceedings of the Workshop on Language Technologies for Digital Humanities and Cultural Heritage*, pages 32–42, Hissar, Bulgaria. Association for Computational Linguistics.

Marcel Bollmann, Anders Søgaard, and Joachim Bingel. 2018. Multi-task learning for historical text normalization: Size matters. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 19–24, Melbourne, Australia. Association for Computational Linguistics.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Bryan Jurish. 2010a. Comparing Canonicalizations of Historical German Text. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 72–77, Uppsala, Sweden. Association for Computational Linguistics.

Bryan Jurish. 2010b. More Than Words: Using Token Context to Improve Canonicalization of Historical German. *Journal for Language Technology and Computational Linguistics*, 25:23–40.

Vladimir. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet physics doklady*, 10(8).

Eva Pettersson. 2016. *Spelling Normalisation and Linguistic Analysis of Historical Text for Information Extraction*. Ph.D. thesis, Uppsala Universitet.

Eva Pettersson, Beáta Megyesi, and Joakim Nivre. 2013. Normalisation of Historical Text Using Context-Sensitive Weighted Levenshtein Distance and Compound Splitting. In *Proceedings of the 19th Nordic Conference of Computational Linguistics*, page 17, Oslo, Norway. Linköping University Electronic Press, Sweden.

Gerold Schneider, Eva Pettersson, and Michael Percillier. 2017. Comparing Rule-based and SMT-based Spelling Normalisation for English Historical Texts. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pages 40–46, Gothenburg. Linköping University Electronic Press.

Caroline T. Schroeder and Amir Zeldes. 2013. Coptic SCRIPTORIUM.

Gael Vaamonde, Luisa Costa, Rita Marquilhas, Clara Pinto, and Fernanda Pratas. 2014. Post Scriptum: Archivo Digital de Escritura Cotidiana. *Janus: Estudos sobre el Siglo de Oro*, pages 473–482.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, Long Beach, CA. Curran Associates, Inc.

Amir Zeldes and Caroline T. Schroeder. 2016. An NLP Pipeline for Coptic. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 146–155, Berlin, Germany. Association for Computational Linguistics.