From the Paft to the Fiiture: a Fully Automatic NMT and Word Embeddings Method for OCR Post-Correction

Mika Hämäläinen♠ Simon Hengchen♢

♠ Department of Digital Humanities, University of Helsinki
♢ COMHIS, University of Helsinki
firstname.lastname@helsinki.fi

Abstract

A great deal of historical corpora suffer from errors introduced by the OCR (optical character recognition) methods used in the digitization process. Correcting these errors manually is a time-consuming process and a great part of the automatic approaches have been relying on rules or supervised machine learning. We present a fully automatic unsupervised way of extracting parallel data for training a character-based sequence-to-sequence NMT (neural machine translation) model to conduct OCR error correction.

1 Introduction

Historical corpora are a key resource to study social phenomena such as language change in a diachronic perspective. Approaching this from a computational point of view is especially challenging as historical data tends to be noisy. The noise can come from OCR (optical character recognition) errors, or from the fact that the spelling conventions have changed as the time has passed, as thoroughly described by Piotrowski (2012).

However, depending on the NLP or DH task being modelled, some methods can cope with the noise in the data. Indeed, Hill and Hengchen (2019) use a subset of an 18th-century corpus, ECCO,[1] and its ground truth version, ECCO-TCP,[2] to compare the output of different common DH methods such as authorship attribution, count-based vector space models, and topic modelling, and report that those analyses produce statistically similar output despite noisiness due to OCR. Their conclusion is similar to Rodriquez et al. (2012) in the case of NER and to Franzini et al. (2018) in the case of authorship attribution, but different from Mutuvi et al. (2018) who, specifically on topic modelling for historical newspapers, confirm the often repeated trope of data too dirty to use. However, reducing the noise of OCRed text by applying a post-correction method makes it possible to gain the full potential of the data without having to re-OCR it and opens up the possibility to process it with the myriad of more precise NLP tools designed for OCR-error free text.

This paper focuses on correcting the OCR errors in ECCO. We present an unsupervised method based on the advances neural machine translation (NMT) in historical text normalization[3]. As NMT requires a parallel dataset of OCR errors and their corresponding correct spellings, we propose a method based on word embeddings, a lemma list, and a modern lemmatizer to automatically extract parallel data for training the NMT model.

2 Related Work

OCR quality for historical texts has recently received a lot of attention from funding bodies and data providers. Indeed, Smith and Cordell (2019) present a (USA-focused) technical report on OCR quality, and aim to spearhead the efforts on setting a research agenda for tackling OCR problems. Other initiatives such as Adesam et al. (2019) set out to analyse the quality of OCR produced by the Swedish language bank Språkbanken, Drobac et al. (2017) correct the OCR of Finnish newspapers using weighted finite-state methods, Tanner et al. (2009) measure mass digitisation in the context of British newspaper archives, while the Euro-

[1] Eighteenth Century Collections Online (ECCO) is a dataset which "contains over 180,000 titles (200,000 volumes) and more than 32 million pages", according to its copyright holder Gale: https://www.gale.com/primary-sources/eighteenth-century-collections-online.

[2] ECCO-TCP (Text Creation Partnership) "is a keyed subset of ECCO, compiled with the support of 35 libraries and made up of 2,231 documents". (Hill and Hengchen, 2019)

[3] Our code https://github.com/mikahama/natas

Proceedings of Recent Advances in Natural Language Processing, pages 431–436,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_051

pean Commission-funded IMPACT project[4] gathers 26 national libraries and commercial providers to "take away the barriers that stand in the way of the mass digitization of the European cultural heritage" by improving OCR technology and advocating for best practices.

Dong and Smith (2018) present an unsupervised method for OCR post-correction. As opposed to our character-level approach, they use a word-level sequence-to-sequence approach. As such a model requires training data, they gather the data automatically by using repeated texts. This means aligning the OCRed text automatically with matched variants of the same text from other corpora or within the OCRed text itself. In contrast, our unsupervised approach does not require any repetition of text, but rather repetition of individual words.

Different machine translation approaches have been used in the past to solve the similar problem of text normalization, which means converting text written in a non-standard form of a language to the standard form in order to facilitate its processing with existing NLP tools. SMT (statistical machine translation) has been used previously, for instance, to normalize historical text (Pettersson et al., 2013) to modern language and to normalize modern Swiss German dialects (Samardzic et al., 2015) into a unified language form. More recently with the rise of the NMT, research has emerged in using NMT to normalize non-standard text, for example work on normalization of medieval German (Korchagina, 2017) and on historical English (Hämäläinen et al., 2018).

All of the normalization work cited above on using machine translation for normalization has been based on character-level machine translation. This means that words are split into characters and the translation model will learn to translate from character to character instead of word to word.

3 Model

As indicated by the related work on text normalization, character-level machine translation is a viable way of normalizing text into a standard variety. Therefore, we will also use character-level NMT in building our sequence-to-sequence OCR post-correction model. However, such a model requires parallel data for training. First, we will present our method of automatically extracting parallel data from our corpus containing OCR errors, then we will present the model designed to carry out the actual error correction.

3.1 Extracting Parallel Data

To extract a parallel corpus of OCR errors and their correctly spelled counterparts out of our corpus, we use a simple procedure consisting of measuring the similarity of the OCR errors with their correct spelling candidates. The similarity is measured in two ways, on the one hand an erroneous form will share a similarity in meaning with the correct spelling as they are realizations of the same word. On the other hand, an erroneous form is bound to share similarity on the level of characters, as noted by Hill and Hengchen (2019) in their study of OCR typically failing on a few characters on the corpus at hand.

In order to capture the semantic similarity, we use Gensim (Řehůřek and Sojka, 2010) to train a Word2Vec (Mikolov et al., 2013) model.[5] As this model is trained on the corpus containing OCR errors, when queried for the most similar words with a correctly spelled word as input, the returned list is expected to contain OCR errors of the correctly spelled word together with real synonyms, the key finding which we will exploit for parallel data extraction.

As an example to illustrate the output of the Word2Vec model, a query with the word *friendship* yields *friendlhip, friendihip, friendflip, friend-, affection, friendthip, gratitude, affetion, friendflhip* and *friendfiip* as the most similar words. In other words, in addition to the OCR errors of the word queried for, other correctly-spelled, semantically similar words (*friend-, affection* and *gratitude*) and even their erroneous forms (*affetion*) are returned. Next, we will describe our method (as shown in Algorithm 1) to reduce noise in this initial set of parallel word forms.

As illustrated by the previous example, we need a way of telling correct and incorrect spellings apart. In addition, we will need to know which incorrect spelling corresponds to which correct spelling (*affetion* should be grouped with *affection* instead of *friendship*).

For determining whether a word is a correctly spelled English word, we compare it to the lem-

[4] http://www.impact-project.eu

[5] Parameters: CBOW architecture, window size of 5, frequency threshold of 100, 5 epochs. Tokens were lowercased and no stopwords were removed.

mas of the Oxford English Dictionary (OED).[6] If the word exists in the OED, it is spelled correctly. However, as we are comparing to the OED lemmas, inflectional forms would be considered as errors, therefore, we lemmatize the word with spaCy[7] (Honnibal and Montani, 2017). If neither the word nor its lemma appear in the OED, we consider it as an OCR error.

For a given correct spelling, we get the most similar words from the Word2Vec model. We then group these words into two categories: correct English words and OCR errors. For each OCR error, we group it with the most similar correct word on the list. This similarity is measured by using Levenshtein edit distance (Levenshtein, 1966). The edit distances of the OCR errors to the correct words they were grouped with are then computed. If the distance is higher than 3 – a simple heuristic, based on ad-hoc testing –, we remove the OCR error from the list. Finally, we have extracted a small set of parallel data of correct English words and their different erroneous forms produced by the OCR process.

Algorithm 1: Extraction of parallel data

```
Draw words w from the input word list;
for w do
    Draw synonyms s_w in the word
      embedding model
    for synonym s_w do
        if s_w is correctly spelled then
            Add s_w to correct forms forms_c
        end
        else
            Add s_w to error forms forms_e
        end
    end
    for error e in forms_e do
        group e with the correct form in
          forms_c by Lev_min
        if Lev_(e,c) > 3 then
            remove(e)
        end
    end
end
```

We use the extraction algorithm to extract the parallel data by using several different word lists. First, we list all the words in the vocabulary of the Word2Vec model and list the words that are correctly spelled. We use this list of correctly spelled words in the model to do the extraction. However, as this list introduces noise to the parallel data, we combat this noise by producing another list of correctly spelled words that have occurred over 100,000 times in ECCO. For these two word lists, one containing all the correct words in the model and the other filtered with word frequencies, we produce parallel datasets consisting of words longer or equal to 1, 2, 3, 4 and 5. The idea behind these different datasets is that longer words are more likely to be matched correctly with their OCR error forms, and also frequent words will have more erroneous forms than less frequent ones.

source	all	>=2	>=3	>=4	>=5
W2V all	29013	28910	27299	20732	12843
W2V freq >100,000	11730	11627	10373	7881	5758
BNC	7692	7491	6681	5926	4925

Table 1: Sizes of the extracted parallel datasets

In addition, we use the frequencies from the British National Corpus (The BNC Consortium, 2007) to produce one more dataset of words occurring in the BNC over 1000 times to test whether the results can be improved with frequencies obtained from a non-noisy corpus. This BNC dataset is also used to produce multiple datasets based on the length of the word. The sizes of these automatically extracted parallel datasets are shown in Table 1.

3.2 The NMT Model

We use the automatically extracted parallel datasets to train a character level NMT model for each dataset. For this task, we use OpenNMT[8] (Klein et al., 2017) with the default parameters except for the encoder where we use a BRNN (bi-directional recurrent neural network) instead of the default RNN (recurrent neural network) as BRNN has been shown to provide a performance gain in character-level text normalization (Hämäläinen et al., 2019). We use the default of two layers for both the encoder and the decoder and the default attention model, which is the general global attention presented by Luong et al. (2015). The models are trained for the default number of 100,000 training steps with the

[6] http://www.oed.com.

[7] Using the `en_core_web_md model`.

[8] Version 0.2.1 of opennmt-py

source	all			>=2			>=3			>=4			>=5		
	Correct	False positive	No output	Correct	False positive	No output	Correct	False positive	No output	Correct	False positive	No output	Correct	False positive	No output
W2V all	0,510	0,350	0,140	0,500	0,375	0,125	0,520	0,325	0,155	0,490	0,390	0,120	0,525	0,390	0,085
W2V freq >100,000	0,515	0,305	0,180	0,540	0,310	0,150	0,510	0,340	0,150	0,540	0,315	0,145	0,515	0,330	0,155
BNC	**0,580**	0,285	0,135	0,555	0,300	0,145	0,570	**0,245**	0,185	0,550	0,310	0,140	0,550	0,315	0,135

Table 2: Results of the NMT models trained on different datasets

same seed value.

We use the trained models to do a character level translation on the erroneous words. We output the top 10 candidates produced by the model, go through them one by one and check whether the candidate word form is a correct English word (as explained in section 3.1). The first candidate that is also a correct English word is considered as the corrected form produced by the system. If none of the top 10 candidates is a word in English, we consider that the model failed to produce a corrected form. The use of looking at the top 10 candidates instead of the topmost candidates is motivated by the findings by Hämäläinen et al. (2019) in historical text normalization with a character-level NMT.

4 Evaluation

For evaluation, we prepare by hand a gold standard containing 200 words with OCR errors from the ECCO and their correct spelling. The performance of our models calculated as a percentage of how many erroneous words they were able to fix correctly. As opposed to the other common metrics such as character error rate and word error rate, we are measuring the absolute performance in predicting the correct word for a given erroneous input word.

Table 2 shows the results for each dataset. The highest accuracy of 58% is achieved by training the model with all of the frequent words in the BNC, and the lowest number of false positives (i.e. words that do exist in English but are not the right correction for the OCR error) is achieved by the model trained with the BNC words that are at least 3 characters long. The *No output* column shows the number of words the models didn't output any word for that would have been correct English.

If, instead of using NMT, we use the Word2Vec extraction method presented in section 3.1 to conduct the error correction by finding the semantically similar word with the lowest edit distance under 4 for an erroneous form, the accuracy of such a method is only 26%. This shows that training an NMT model is a meaningful part in the correction process.

In the spirit of Hämäläinen et al. (2018), whose results indicate that combining different methods in normalization can be beneficial, we can indeed get a minor boost for the results of the highest accuracy NMT model if we first try to correct with the above described Word2Vec method and then with NMT, we can increase the overall accuracy to 59.5%. However, there is no increase if we invert the order and try to first correct with the NMT and after that with the Word2Vec model.

5 Conclusion and Future Work

In this paper we have proposed an unsupervised method for correcting OCR errors. Apart from the lemma list and the lemmatizer, which can also be replaced by a morphological FST (finite-state transducer) analyzer or a list of word forms, this method is not language specific and can be used even in scenarios with less NLP resources than what English has. Although not a requirement, having the additional information about word frequencies from another OCR error-free corpus can boost the results.

A limitation of our approach is that it cannot do word segmentation in the case where multiple words have been merged together as a result of the OCR process. However, this problem is complex enough on its own right to deserve an entire publication of its own and is thus not in the scope of our paper. Indeed, previous research has been conducted focusing solely on the segmentation problem (Nastase and Hitschler, 2018; Soni et al., 2019) of historical text and in the future such methods can be incorporated as a preprocessing step for our proposed method.

It is in the interest of the authors to extend the approach presented in this paper on historical data written in Finnish and in Swedish in the immediate near future. The source code and the best working NMT model discussed in this paper has be made freely available on GitHub as a part of the natas Python library[9].

[9]https://github.com/mikahama/natas

Acknowledgements

We would like to thank the COMHIS group[10] for their support, as well as GALE for providing the group with ECCO data.

References

Yvonne Adesam, Dana Dannélls, and Nina Tahmasebi. 2019. Exploring the quality of the digital historical newspaper archive kubhist. *Proceedings of DHN*.

Rui Dong and David Smith. 2018. Multi-input attention for unsupervised OCR correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2363–2372.

Senka Drobac, Pekka Sakari Kauppinen, Bo Krister Johan Linden, et al. 2017. OCR and post-correction of historical finnish texts. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*. Linköping University Electronic Press.

Greta Franzini, Mike Kestemont, Gabriela Rotari, Melina Jander, Jeremi K Ochab, Emily Franzini, Joanna Byszuk, and Jan Rybicki. 2018. Attributing authorship in the noisy digitized correspondence of Jacob and Wilhelm Grimm. *Frontiers in Digital Humanities*, 5:4.

Mika Hämäläinen, Tanja Säily, Jack Rueter, Jörg Tiedemann, and Eetu Mäkelä. 2018. Normalizing early English letters to present-day English spelling. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 87–96, Santa Fe, New Mexico. Association for Computational Linguistics.

Mika Hämäläinen, Tanja Säily, Jack Rueter, Jörg Tiedemann, and Eetu Mäkelä. 2019. Revisiting NMT for normalization of early English letters. In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 71–75, Minneapolis, USA. Association for Computational Linguistics.

Mark J. Hill and Simon Hengchen. 2019. Quantifying the impact of dirty OCR on historical text analysis: Eighteenth Century Collections Online as a case study. *Digital Scholarship in the Humanities: DSH*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing. *To appear*.

[10] https://www.helsinki.fi/en/researchgroups/computational-history

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proc. ACL*.

Natalia Korchagina. 2017. Normalizing medieval german texts: from rules to deep learning. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pages 12–17.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, 8, pages 707–710.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Stephen Mutuvi, Antoine Doucet, Moses Odeo, and Adam Jatowt. 2018. Evaluating the impact of OCR errors on topic modeling. In *International Conference on Asian Digital Libraries*, pages 3–14. Springer.

Vivi Nastase and Julian Hitschler. 2018. Correction of OCR word segmentation errors in articles from the ACL collection through neural machine translation methods. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Eva Pettersson, Beáta Megyesi, and Jörg Tiedemann. 2013. An SMT approach to automatic annotation of historical text. In *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013; May 22-24; 2013; Oslo; Norway. NEALT Proceedings Series 18*, 087, pages 54–69. Linköping University Electronic Press.

Michael Piotrowski. 2012. Natural language processing for historical texts. *Synthesis lectures on human language technologies*, 5(2):1–157.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Kepa Joseba Rodriquez, Mike Bryant, Tobias Blanke, and Magdalena Luszczynska. 2012. Comparison of named entity recognition tools for raw OCR text. In *KONVENS*, pages 410–414.

Tanja Samardzic, Yves Scherrer, and Elvira Glaser. 2015. Normalising orthographic and dialectal variants for the automatic processing of Swiss German. In *Proceedings of the 7th Language and Technology Conference*.

David A. Smith and Ryan Cordell. 2019. A research agenda for historical and multilingual optical character recognition. Technical report, Northeastern University.

Sandeep Soni, Lauren Klein, and Jacob Eisenstein. 2019. Correcting whitespace errors in digitized historical texts. In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 98–103, Minneapolis, USA. Association for Computational Linguistics.

Simon Tanner, Trevor Muñoz, and Pich Hemy Ros. 2009. Measuring mass text digitization quality and usefulness. *D-lib Magazine*, 15(7/8):1082–9873.

The BNC Consortium. 2007. The British National Corpus, version 3 (BNC XML Edition). Distributed by Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium. Http://www.natcorp.ox.ac.uk/.