Neural Machine Translation with BERT for Post-OCR Error Detection and Correction

Thi-Tuyet-Hai Nguyen
L3i, University of La Rochelle
hai.nguyen@univ-lr.fr

Adam Jatowt
Kyoto University
adam@dl.kuis.kyoto-u.ac.jp

Nhu-Van Nguyen
L3i, University of La Rochelle
nhu-van.nguyen@univ-lr.fr

Antoine Doucet
L3i, University of La Rochelle
antoine.doucet@univ-lr.fr

Mickael Coustaty
L3i, University of La Rochelle
mickael.coustaty@univ-lr.fr

ABSTRACT

The quality of OCR has a direct impact on information access, and an indirect impact on the performance of natural language processing applications, making fine-grained (e.g., semantic) information access even harder. This work proposes a novel post-OCR approach based on a contextual language model and neural machine translation, aiming to improve the quality of OCRed text by detecting and rectifying erroneous tokens. This new technique obtains results comparable to the best-performing approaches on English datasets of the competition on post-OCR text correction in ICDAR 2017/2019.

KEYWORDS

post-OCR processing, BERT, neural machine translation

1 INTRODUCTION

Historical documents contain valuable knowledge that gets considerable attention from researchers and libraries around the world. Substantial efforts have been devoted to transforming paper-based documents into electronic text in order to preserve them as well as make them fully accessible.

Limitations of modern OCR technologies in handling historical documents lead to difficulties in reading, retrieving as well as other processes on digitized collections [17]. In other words, they reduce the benefits of digitization projects by making it difficult for users to acquire knowledge from past documents. Our work attempts at minimizing the influences of the OCR problems by detecting and correcting errors of digitized texts. Bidirectional encoder representations from transformers (BERT) and neural machine translation (NMT) are employed in our approach with some variations.

A shared task is a good chance to compare techniques. Therefore, we use the evaluation metrics and English datasets of the two occurrences of the competition on post-OCR text correction in 2017 [3] and 2019 [14] to evaluate the performance of our proposed methods. Experimental results show that our approach performs slightly better than the winners of the competition on error detection and obtains comparable improvements on error correction.

Our three contributions are mentioned as follows. The first one is to apply static word embeddings in fine-tuned BERT models, which increases performance of error detection. Our character embeddings created by training NMT on aligned OCRed text and its ground truth (GT) achieve some positive results in rectifying errors. The last contribution is to utilize a length difference for removing irrelevant candidates, which improves correction output.

2 RELATED WORK

The literature of OCR post-processing research has a rich family of models. They are grouped into three types: manual approach type which lets human manually review and correct OCRed texts, lexical approach type based on the comparison of source words to a dictionary entries, and statistical approach type that utilizes error distributions from training data.

The manual type. Crowd-sourcing (e.g. [6]) is one of key approaches of the manual type. While the collaborative OCR correction approaches work effectively with high accuracy, they also have some limitations. They require the original documents which are often unavailable for some OCRed corpora. In addition, these methods heavily depend on volunteer work.

The lexical type. The approaches of the lexical type typically exploit distance measures between an erroneous word and a lexicon entry to suggest candidates for correcting OCRed errors.

Some prior work (e.g [15]) studies the influence of a lexicon coverage and different ways of dynamically collecting specialized lexicons. Instead of building a dictionary, Bassil *et al.* [2] harness Google's massive indexed data for post-processing OCR output. One of competition teams (EFP) [3] also explores lexicon look-up techniques and regular expressions to detect and correct errors.

Lexical approach type is easy to be applied, however, it also goes together with some difficulties. Historical documents do not follow the same spelling rules as modern texts and often lack complete lexicons. Moreover, the approaches of this type only concentrate on single words so they cannot tackle real-word errors (e.g. 'hear' is a real-word error in a phrase 'stay hear in Japan').

The statistical type. Most of the post-processing approaches are statistical, which enable to model specific distributions of the target domain from available training data.

Some methods (e.g. [9]) combine different digitized outputs of the same paper-based document to benefit from each other. Some approaches (e.g. [5, 7, 11], 2-pass RNN, CSIITJ, RAE or WFST-PostOCR - competition teams [3, 14], etc.) employ error model and language model in various ways to detect and correct remaining erroneous tokens. Others (e.g. Char-SMT/NMT [1], MMDT [16], CLAM, CCC, UVA - competition teams [3, 14]) use machine translation techniques in order to transform OCRed text into corrected one.

In the 2017 and 2019 competitions on post-OCR text correction [3, 14], participants implemented various methods to detect and correct OCRed errors. The best-performing of the detection task was the fine-tuned BERT model of CCC team. Both of the winners in the correction task (Char-NMT/SMT, CCC) use character-level machine translation techniques at character level with some additional features. Their methods outperform other approaches based on error model and language model, or those of the lexical type.

Consequently, our proposal is developed from BERT and character-level machine translation model with some extensions, including static embeddings applied in BERT, our character embeddings used in NMT, and candidate filter.

3 ERROR DETECTION

BERT [4] is a multi-layer bidirectional transformer encoder. It is pre-trained on unlabeled data over two different tasks, including Masked Language Model (MLM), and Next Sentence Prediction (NSP). BERT models can be fine-tuned to handle NLP problems. Downstream tasks are firstly set with the pre-trained parameters, which are adjusted by their labelled data.

There are multiple task-specific BERT models [4], some of them work at sentence level, others perform at token level. Error detection problem can be viewed as token classification which classifies OCRed tokens as either valid or invalid. We focus on fine-tuning BERT models at token level.

We adapt the model of named entity recognition (NER) to an error detection model. Particularly, instead of tagging tokens with NER taggers, we tag tokens with label 1 (invalid token) or 0 (valid token). Our approach is similar to the one of the winner of the 2019 competition, but we simplify the model with only one fully-connected layer on the top of the hidden-states output. In addition, it is proved that pre-trained word embedding models increase the performance of NLP tasks. Thus, instead of randomly initializing embeddings like the competition winner CCC does, we employ popular word embeddings (Fasttext, Glove) in our model.

Our approach consists of the four following steps. OCRed input is first split into OCRed tokens based on white-space. Next, we apply WordPiece [18] tokenization on each token to get corresponding sub-tokens. A mapping between the original OCRed token and its sub-tokens is also maintained. Then, Glove or Fasttext is used to embed sub-tokens in lieu of assigning random numbers as initial embeddings.

After that, these embeddings are combined with segment and position embeddings as inputs of BERT token classification model, which is a BERT model with an additional fully-connected layer. This design is simpler than the state of the art which uses both convolutional and fully-connected layers. The outcome of this stage is labelled sub-tokens, with label 1 for invalid tokens and 0 for valid tokens. Finally, the original tokens are considered as invalid ones if at least one of their sub-tokens is labelled as error.

Take an OCRed sequence 'we wyll go' with an error 'wyll' as an example to illustrate our approach. The input of the first step is a list of OCRed tokens tokenized by white-spaces, {'we', 'wyll', 'go'}. Applying WordPiece on each OCRed token, we have the corresponding sub-tokens and their mappings to their original tokens, {'we': 'we', 'wyll': {'w', '##yl', '##l'}, 'go': 'go'}. Next, the pre-trained word embeddings Glove or Fasttext embed the sub-tokens to be used as inputs for BERT token classification. The classifier labels each sub-token as either a valid or invalid word. The original token ('wyll') is identified as the error since its sub-tokens are classified as invalid ones ('w', '##yl', '##l').

Table 1: Example of input/output sequences

OCRed text (source side)
t w e n t y # i n # n u m b e r # a n d j u s t # t h e n
i n # n u m b e r # a n d j u s t # t h e n # p u b l i s h e d
n u m b e r # a n d j u s t # t h e n # p u b l i s h e d # i n
a n d j u s t # t h e n # p u b l i s h e d # i n # a
GT text (target side)
t w e n t y # i n # n u m b e r # a n d $ j u s t # t h e n
i n # n u m b e r # a n d $ j u s t # t h e n # p u b l i s h e d
n u m b e r # a n d $ j u s t # t h e n # p u b l i s h e d # i n
a n d $ j u s t # t h e n # p u b l i s h e d # i n # a

4 ERROR CORRECTION

As mentioned in Section 2, character-level MT is the state of the art for the error correction task, where it enables to tackle the problem of data sparsity. Regarding MT techniques, SMT consists of many small sub-components that are tuned separately. In contrast, NMT aims at building a single neural network which maximizes the translation performance. Its performance is comparable to the existing state-of-the-art phrase-based model [18]. Consequently, we employ NMT at character level to translate OCRed text into its corrected version (in the same language).

Our models are built on an open-source toolkit for neural machine translation (OpenNMT) [8]. We use most of the default values of OpenNMT, except for embedding, hidden layer size, sequence length. Input and output texts are written in the same language, therefore we configure to share embeddings between the source and target side with embedding size of 160 (tested against 100). Hidden layer size is increased from 500 to 1000 in order to learn more information. We set the maximum sequence length to 70 (instead of the default one, 50) to cover longer sequences of training data.

It is the fact that most of OCRed tokens are correct. If the MT system is trained on a dataset with a large proportion of valid tokens, then it might not rectify errors. In order to reduce the negative impact of imbalanced data and deal with real-word errors, we consider erroneous OCRed tokens and some nearby tokens (which can be correct or incorrect) as input; the corresponding GT texts are provided as output of NMT models.

Particularly, given one error and its four neighbors, we generate five word 5-grams which are represented at character level and used as input sequences. By doing this, we augment data for training NMT models. In the data representation, space and '#' are viewed as character delimiters and word boundary markers, respectively. If an error is a run-on one, '$' is used as word delimiter within its target text. It should be noted that we do not consider an input sequence with all four words on the left side of the error and no word on its right side. The reason is that we expect to tackle incorrect split errors, such as 'main tain' vs. GT word 'maintain'.

For example, given an error 'andjust' in OCRed phrase 'twenty in number andjust then published in a', and its corresponding GT 'twenty in number and just then published in a', four input sequences of the error and their output ones are shown in Table 1.

Furthermore, Sennrich *et al.* concluded that linguistic features (e.g. POS tags, morphological features, etc.) yield high performance of NMT systems. However, these features are specifically designed for words rather than characters. Amrhein *et al.* [1] applied two features in their NMT models, including the text types and the written time span. Nevertheless, both of the features are missing

Table 2: Example of an input sequence of Monograph dataset along with the feature document source M (Monograph).

<table>
<tr><td>OCRed text (source side)</td></tr>
<tr><td>n|M u|M m|M b|M e|M r|M #|M a|M n|M d|M j|M u|M s|M t|M</td></tr>
<tr><td>GT text (target side)</td></tr>
<tr><td>n u m b e r # a n d $ j u s t</td></tr>
</table>

from Comp2019 dataset. We think that OCRed texts of Comp2019 dataset might share some common characteristics, thus, our work considers the source of this dataset as its type. In total, there are three text types in the competition datasets (monograph and periodical from Comp2017, and Comp2019), which are exploited as additional input feature (or factor) for MT model.

By applying factored NMT, we have more training data. Moreover, instead of training different models for each dataset, we only need to train a single model to test on our three datasets. An example of an input sequence of Monograph dataset with factored representation is shown in Table 2. Factored NMT model is the first version of our approach (denoted as Correction 1).

MT techniques apply pre-trained word embeddings to improve translation performance. Several word embeddings are available and free to access while it is not easy to find a character embedding. McCann *et al.* [10] reported that a pre-trained encoder of a MT model increases the performance of other NLP tasks. Their contextualized word vectors are known as Context Vectors (CoVe). Broadening this idea, we extract embeddings from character-level NMT model trained with an aligned data.

Particularly, we align OCRed text with its corresponding GT text, then we generate input sequences from each aligned error with its contextual tokens. New character embeddings are extracted from models trained with the aligned data and shared embeddings between source and target side. It is expected that the embeddings (called as *aligned embeddings*) are able to put characters closer together in the vector space provided that they have similar contexts and/or shapes. The second version of our approach (called as Correction 2) is similar to the first one but uses *aligned embeddings*.

According to previous work [13], more than 80% of OCRed errors have an edit distance less than 3. We apply this feature to remove some irrelevant candidates. Specifically, after getting candidates for each error from MT models, we only select candidates which have edit distance with the error lower than 3. Furthermore, the analyses also indicate that percentage of deletion and insertion errors are much lower than that of substitution errors. While it is expensive to compute edit distance between two sequences, the length difference between candidate length and OCRed token length is simple and fast to calculate. We find that by setting the *length difference* threshold to 4, we obtain a performance comparable to using edit distance. The last version of our approach (denoted as Correction 3) is the same as Correction 2 with the addition of the *length difference* filter.

5 EXPERIMENTAL RESULTS

5.1 Metrics

For the detection task, we use the official metrics of the competition to evaluate our approach: Precision, Recall, F-score. Regarding the correction task, the improvement percentage is used to evaluate compared approaches. This metric is computed based on the difference of the original distance (between GT and OCRed text) and the corrected distance (between GT and the corrected text) which considers the confidence of each candidate to be the correction in case of many candidates for the same error.

Table 3: Details of the evaluation datasets

Dataset	Source	Type	Dates	CER(%)	# Char.
Monograph	BL Monog	mono.	1858 - 1891	1	1.2M
	GT BnF Eng	mono.	1802 - 1911	2	3M
Periodical	BL Euro NP	peri.	1744 - 1894	4	1.8M
Comp2019	IMPACT	-	-	21.28	0.24M

5.2 Datasets

English OCRed texts of both rounds of the competition are exploited as evaluation data. The dataset of Comp2017 consists of 813 English written files that were either published in periodicals or monographs. Therefore, they were divided by the competition organizers into two datasets: Monograph and Periodical. The dataset of Comp2019 contains 200 files in English, which are from IMPACT project. The corresponding GT is created by different projects such as Europeana Newspapers, IMPACT, Project Gutenberg, Perseus and Wikisource. These datasets are distributed as a training set of 80% and an evaluation set of 20%. The detailed characteristics of the used evaluation datasets are shown in Table 3.

5.3 Results

Tables 4 and 5 illustrate the performance of the proposed detection and correction approaches on the competition datasets in ICDAR 2017, ICDAR 2019, respectively.

Error Detection. In overall, our approach surpasses other approaches on Periodical (with 4% higher F-score) and Comp2019 (with 1% higher F-score) but not on Monograph. These results are partly explained by the rate of real-word and non-word errors in each dataset and the strength of our neural network based approach. In fact, there are more real-word errors in two datasets (Periodical and Comp2019) than in Monograph. BERT is a contextual language model, so it is reasonable that the BERT-based model can detect more real-word errors.

The rate of correctly detected real-word errors supports our assumption. Our approach is able to identify 64% of context-sensitive errors on Monograph, 63% on Periodical, 48% on Comp2019, which is better than the results reported in the prior work [12] (43% on Monograph, 49% on Periodical, no report on Comp2019). This approach also gets higher results of correctly detected non-word errors with 95% on Periodical, 93% on Comp2019, but not on Monograph (82%). Similarly, the percentage of correctly recognized OOV words is comparable to the ones reported in the related work [12] with about 61% on average on three datasets.

Error Correction. There is an important difference between the 2017 and 2019 competitions. While the organisers provide the predefined list of error positions, the correction task of the second round is more challenging as it requires not only to identify error positions but also to rectify such detected errors.

In general, the best model of our approach outperforms some of our counterparts. In terms of the 2017 competition, our single model performs better than most of participants, except for the

Table 4: Results on the English datasets of the competition in ICDAR2017 (F-score: detection, Impr.: correction); '-' marks no improvement, 'x' denotes no reported result by the competition or the prior approaches that only work on detection [12] or correction task [11].

	Monograph		Periodical	
Approaches	F-score(%)	Impr.(%)	F-score(%)	Impr.(%)
Char-SMT/NMT [1]	67	**43**	64	**37**
CLAM [3]	67	29	x	22
EFP [3]	69	13	54	-
MMDT [16]	66	20	44	-
WFST-PostOCR [3]	73	28	68	-
2-pass RNN [3]	66	x	66	-
Nguyen *et al.* [12]	**79**	x	70	x
Nguyen *et al.* [11]	x	30	x	10
Detection	72	x	**74**	x
Correction 1	x	31	x	19
Correction 2	x	32	x	20
Correction 3	x	36	x	27

Table 5: Results on the English dataset of the competition in ICDAR 2019 (F-score: detection, Impr.: correction); 'x' marks no reported result by the competition or the past approaches that only work on detection or correction task.

	Comp2019	
Approaches	F-score(%)	Impr.(%)
CCC [14]	**67**	**11**
CLAM [14]	45	0.4
CSIITJ [14]	45	2
RAE1 [14]	53	9
RAE2 [14]	57	6
UVA [14]	47	0
Detection	**68**	x
Correction 1	x	1
Correction 2	x	2
Correction 3	x	4

state-of-the-art approach (Char-SMT/NMT) which combines five different models of statistical MT and neural MT. The authors of Char-SMT/NMT claimed that their system is complicated and difficult to apply to new datasets. Therefore, they suggested the most promising single system [1] which works across all data sets. However, its performance is significantly lower than the ensemble model as well as our proposals. In contrast, our model is easy to implement with available data. Moreover, it should be emphasized that our improvement is much higher than the neural MT based approach (CLAM) or statistical MT based one (MMDT) [16]. Consequently, we think that our model can be considered as a reliable solution to reduce OCR errors across various data sets.

In terms of the 2019 competition (i.e., without the provided error list) we have to use our list of errors obtained from the detection task for generating data sequences. Our best model still underperforms some other methods, including RAE2, RAE1 and CCC. In our opinion, the reason is that our models are built on the limited resources of the 2019 competition which is small and contains several real-word errors involving wrong line recognition. The RAE1, RAE2 competitors and the CCC team benefit from using external materials like the Google Books Ngram Corpus. Nonetheless, there is no clear conclusion between the performance of our best model and that of RAE1&2 (called WFST-PostOCR) as the former performed better in the first round.

6 CONCLUSIONS

This paper presents a novel approach to improve the quality of digitized outputs. Our error detector enables to detect several real-word errors by exploiting word embeddings and pre-trained BERT models. Our correction approach which applies NMT techniques on contextual input data and some additional features is promising to reduce OCRed errors. Nevertheless, if real-word errors relate to wrong line recognition, the performance of our approach is limited. Future work will focus on employing additional external resources to improve our results.

REFERENCES

[1] Chantal Amrhein and Simon Clematide. 2018. Supervised OCR Error Detection and Correction Using Statistical and Neural Machine Translation Methods. *Journal for Language Technology and Computational Linguistics* (2018).

[2] Youssef Bassil and Mohammad Alwani. 2012. Ocr post-processing error correction algorithm using google online spelling suggestion. *Journal of Emerging Trends in Computing and Infor- mation Sciences* (2012).

[3] Guillaume Chiron, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2017. ICDAR2017 competition on post-OCR text correction. In *14th IAPR International Conference on Document Analysis and Recognition*. IEEE, 1423–1428.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] John Evershed and Kent Fitch. 2014. Correcting noisy OCR: Context beats confusion. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*. ACM, 45–51.

[6] Rose Holley. 2009. *Many hands make light work: Public collaborative OCR text correction in Australian historic newspapers*. National Library of Australia.

[7] Ido Kissos and Nachum Dershowitz. 2016. OCR error correction using character correction and feature-based word classification. In *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*. IEEE, 198–203.

[8] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*. 67–72.

[9] William B Lund, Douglas J Kennard, and Eric K Ringger. 2013. Combining multiple thresholding binarization values to improve OCR output. In *Document Recognition and Retrieval XX*. International Society for Optics and Photonics.

[10] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*. 6294–6305.

[11] Thi-Tuyet-Hai Nguyen, Mickaël Coustaty, Antoine Doucet, Adam Jatowt, and Nhu-Van Nguyen. 2018. Adaptive Edit-Distance and Regression Approach for Post-OCR Text Correction. In *Maturity and Innovation in Digital Libraries - 20th International Conference on Asia-Pacific Digital Libraries, ICADL 2018*. 278–289.

[12] Thi-Tuyet-Hai Nguyen, Mickaël Coustaty, Antoine Doucet, Adam Jatowt, and Nhu-Van Nguyen. 2019. Post-OCR Error Detection by Generating Plausible Candidates. In *15th IAPR International Conference on Document Analysis and Recognition, ICDAR 2019*.

[13] Thi-Tuyet-Hai Nguyen, Adam Jatowt, Mickaël Coustaty, Nhu-Van Nguyen, and Antoine Doucet. 2019. Deep Statistical Analysis of OCR Errors for Effective Post-OCR Processing. In *19th ACM/IEEE Joint Conf. on Digital Libraries*. 29–38.

[14] Christophe Rigaud, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2019. ICDAR 2019 Competition on Post-OCR Text Correction. In *15th IAPR International Conference on Document Analysis and Recognition, ICDAR 2019*.

[15] Christoph Ringlstetter, Klaus U Schulz, and Stoyan Mihov. 2007. Adaptive text correction with Web-crawled domain-dependent dictionaries. *ACM Transactions on Speech and Language Processing (TSLP)* 4, 4 (2007), 9.

[16] Sarah Schulz and Jonas Kuhn. 2017. Multi-modular domain-tailored OCR post-correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2716–2726.

[17] Myriam C Traub, Jacco Van Ossenbruggen, and Lynda Hardman. 2015. Impact analysis of ocr quality on research tasks in digital archives. In *International Conference on Theory and Practice of Digital Libraries*. Springer, 252–263.

[18] Yonghui Wu, Mike Schuster, Zhifeng Chen, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).