Text-cleaning - OCR post-processing text correction

Shripad Ambure, INDIA

Author
Shripad Ambure(MSc),
Department of Computer Science,
Universita degli studi di Milano,
MILAN, 20122 , ITALY

Email:
shripad.ambure@studenti.unimi.it

OCR has been a busy study topic for over the last 30 but findings are still poor, particularly for historical documents. The goal of this project is to compare and evaluate automated systems for repairing OCR-ed documents. The current challenge consists of two tasks: 1)error detection and 2) error correction.OCR quality has a direct influence on information access and an indirect impact on the performance of natural language processing applications, making fine-grained (e.g., semantic) information access even more difficult. This paper offers a unique post-OCR technique based on a contextual language model and neural machine translation, with the goal of improving OCRed text quality by identifying and correcting erroneous tokens. This novel strategy achieves results similar to the best-performing algorithms on datasets from the ICDAR 2019 competition on post-OCR text repair.

KEYWORDS
OCR-Post Processing, text correction, BERT Embeddings, Neural Machine Translation

1 | INTRODUCTION

Researchers and libraries all across the globe are interested in historical records because they provide significant information.To preserve paper-based records while still making them fully accessible, much work has been put into converting them to electronic text.Due to the limitations of present OCR technology in handling ancient texts, reading, retrieving, and other activities on digital collections are challenging. To put it another way, they limit the advantages of digitization efforts by making it impossible for users to learn from old materials. Our research aims to reduce the negative effects of OCR issues by identifying and repairing faults in digital texts. With certain modifications, we use

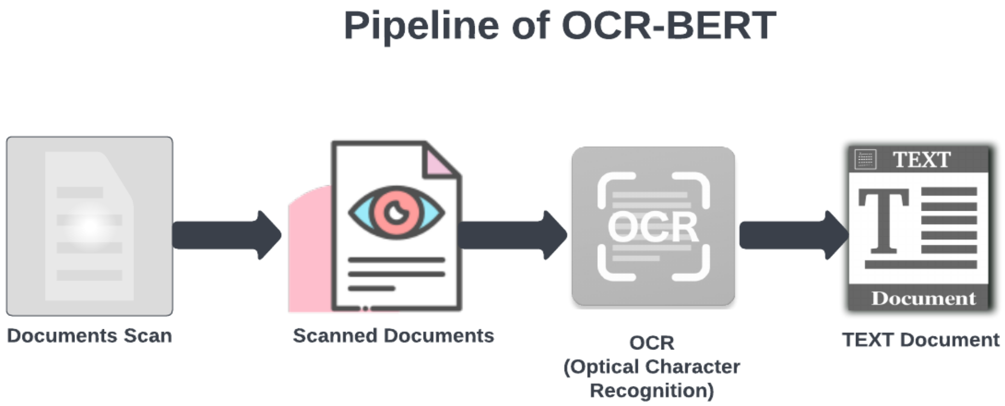


FIGURE 1 Pipeline of OCR-BERT

bidirectional encoder representations from transformers (BERT) and neural machine translation (NMT) in our method. A common activity provides an excellent opportunity to compare methods. To test the performance of our proposed methods, we utilise the evaluation measures and English datasets from the two editions of the competition on post-OCR text correction in 2017 and 2019. Experiments demonstrate that our technique detects errors somewhat better than the competition's winners and achieves equivalent gains in mistake rectification. The following is a list of our three contributions. The first is to use static word embeddings in fine-tuned BERT models to improve error detection performance. Our character embeddings, which were constructed by training NMT on aligned OCRed text and its ground truth (GT), show some promise in terms of mistake correction. The last contribution is to remove unnecessary candidates using a length difference, which enhances rectification output[1].

1.1 | Related Work

Errors in unedited writings can fall into several types, depending on their source: erroneous word segmentation, typos, purposeful misspellings, including shortened/phonetically written words (particularly on social media),characters, historical documents with non-canonical spellings, grammatical faults, and possibly more.Neural-based solutions can be used to solve many of the challenges stated above.techniques designed originally for machine translation have shown to be really effective. Yannakoudakis and colleagues N-best (2017) uses a machine translation-inspired technique. Using neural sequence labelling models to rank lists –rectification of grammatical errors[2].

There are several models in the OCR post-processing literature. They are divided into three categories: manual technique, which allows humans to manually evaluate and fix OCRed documents; lexical approach, which compares source words to dictionary entries; and statistical approach, which uses error distributions from training data.

Manual Method - One of the most important manual ways is crowd-sourcing. While collaborative OCR correction methods perform well and provide high accuracy, they do have significant drawbacks. They require original documents, which are frequently unavailable in OCRed corpora. Furthermore, these systems rely substantially on volunteer effort.

lexical Method - To identify possibilities for repairing OCRed mistakes, lexical techniques often use distance measurements between an incorrect term and a lexicon item.Previous research has looked at the impact of lexicon coverage and different methods for dynamically obtaining specialised lexicons. Although the lexical method is simple to implement, it does come with certain challenges. Historical manuscripts frequently lack entire lexicons and may not follow the same spelling conventions as current literature.

Statistical Method - The majority of post-processing methods are statistical, allowing for the modelling of specified target domain distributions using existing training data. To benefit from each other, several approaches integrate different digital outputs of the same paper-based document. To convert OCRed text into corrected text, we apply machine translation algorithms. Participants in the post-OCR text correction contests in 2017 and 2019 used a variety of ways to detect and rectify OCRed mistakes.Our solution, on the other hand, is based on BERT and a character-level machine translation model with various modifications, such as static embeddings in BERT, our character embeddings in NMT, and a candidate filter[3][4].

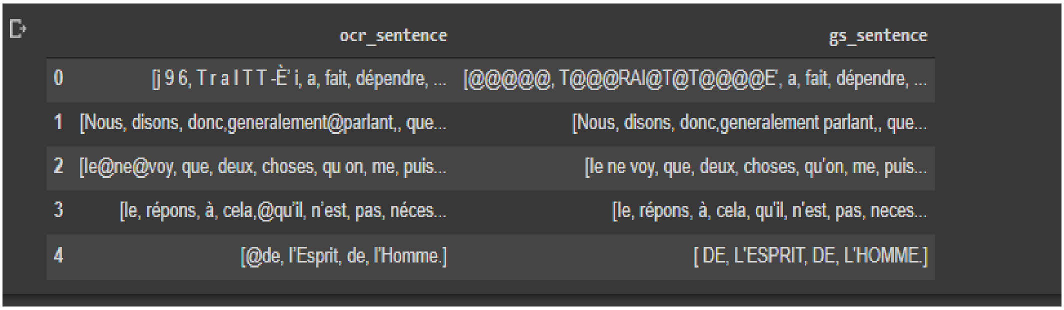
2 | TYPO DETECTION

BERT is a bidirectional multi-layer transformer encoder. Masked Language Model (MLM) and Next Sentence Prediction are two tasks that it has been pre-trained on unlabeled data (NSP). NLP difficulties can be handled by fine-tuning BERT models. The pre-trained parameters are used to establish downstream tasks, which are then changed by their labelled data. There are a variety of task-specific BERT models available; some function at the phrase level, while others work at the token level. Token classification, which classifies OCRed tokens as genuine or invalid, may be considered as an error detection task. At the token level, we work on fine-tuning BERT models.

The named entity recognition (NER) paradigm is adapted to an error detection model. Specifically, instead of using NER taggers to tag tokens, we use labels 1 (invalid token) and 0 (valid token). Our strategy is similar to that of the 2019 competition winner, but we simplify the model by adding only one fully connected layer on top of the hidden-states output. Furthermore, it has been demonstrated that pre-trained word embedding models improve NLP task performance. We use common word embeddings (Fasttext, Glove) in our model instead of randomly initialising embeddings, as the competition winner CCC does. The four steps that comprise our strategy are as follows. On the basis of white-space, OCRed input is separated into OCRed tokens. To retrieve appropriate sub-tokens, we use Word Piece [18] tokenization to each token. There is also a mapping between the original OCRed token and its sub-tokens. Then, instead of assigning random integers as starting embeddings, Glove or Fast text is utilised to embed sub-tokens.

These embeddings are then merged with segment and position embeddings as inputs to the BERT token classification model, which is a BERT model with an extra fully-connected layer. This approach is more straightforward than the current state of the art, which incorporates both convolutional and fully-connected layers. This stage's output is labelled sub-tokens, with 1 indicating invalid tokens and 0 indicating valid tokens. Finally, if at least one of the sub-tokens is labelled as incorrect, the original tokens are declared invalid. To demonstrate our technique, consider the OCRed sequence 'we wyll go' with the mistake 'wyll'. The first step's input is a list of OCRed tokens tokenized by white-spaces, such as 'we,' 'wyll,' and 'go.' We have the corresponding sub-tokens and their mappings to their original tokens using WordPiece on each OCRed token, 'we': 'we', 'wyll': 'w', 'yl', 'l', 'go': 'go'. The sub-tokens to be used as inputs for BERT token classification are then embedded using the pre-trained word embeddings Glove or Fasttext. Each sub-token is classified as a valid or invalid word by the classifier. Because its sub-tokens ('w', 'yl', and 'l') are categorised as invalid, the original token ('wyll') is recognised as the mistake.

On a new line, each file comprises OCRed text, aligned OCRed text, and aligned GS text. The length of the aligned OCRed and GS texts is always the same, and the missing letters are shown by the "@" symbol.



	ocr_sentence	gs_sentence
0	[j 9 6, T r a I T T -È' i, a, fait, dépendre, ...	[@@@@@, T@@@RAI@T@T@@@@E', a, fait, dépendre, ...
1	[Nous, disons, donc,generalement@parlant,, que...	[Nous, disons, donc,generalement parlant,, que...
2	[le@ne@voy, que, deux, choses, qu on, me, puis...	[le ne voy, que, deux, choses, qu'on, me, puis...
3	[le, répons, à, cela,@qu'il, n'est, pas, néces...	[le, répons, à, cela, qu'il, n'est, pas, neces...
4	[@de, l'Esprit, de, l'Homme.]	[DE, L'ESPRIT, DE, L'HOMME.]

FIGURE 2 Example of OCRed sentence and GS sentences

3 | TYPO CORRECTION

As stated in Section 2, character-level MT is the state of the art for error correction tasks, allowing for the solution of data sparsity problems. SMT, in terms of MT methods, is made up of a number of tiny sub-components that are adjusted independently. NMT, on the other hand, tries to create a single neural network that optimises translation performance. Its performance is similar to that of the current best-in-class phrase-based model. As a result, we use NMT at the character level to convert OCRed text into its corrected form (in the same language). Our neural machine translation models are based on an open-source toolset (Open NMT). Except for embedding, hidden layer size, and sequence length, we utilise most of Open NMT's default defaults. Because the input and output texts are written in the same language, we set the source and target sides to share embeddings with a 160-pixel embedding size (tested against 100). In order to gain additional information, the size of the hidden layer has been raised from 500 to 1000. To cover larger sequences of training data, we increased the maximum sequence length to 70 (rather than the default of 50).The reality is that the majority of OCRed tokens are accurate. If the MT system is trained on a dataset with a high percentage of valid tokens, it may not be able to correct mistakes. We use erroneous OCRed tokens and some neighbouring tokens (which might be accurate or wrong) as input and the related GT texts as output of NMT models to limit the detrimental effect of unbalanced data and cope with real-word mistakes.We construct five word 5-grams from one mistake and its four neighbours, which are represented at the character level and utilised as input sequences. By doing so, we can increase the amount of data available for training NMT models. Space and '' are used as character delimiters and word boundary markers, respectively, in the data format. Within the target text of a run-on error, the word delimiter '@' is utilised. It's worth noting that an input sequence with all four words on the left side of the error and no word on the right side isn't taken into account. The rationale for this is because we intend to deal with erroneous split mistakes like'main tain' vs. GS word 'maintain'.We assume that OCRed texts of Comp2019 dataset could have certain similar properties, hence, our study considers the source of this dataset as its type. In all, there are three text kinds in the competition datasets (monograph and periodical from Comp2017, and Comp2019), which are utilised as extra input feature (or factor) for MT model. By using factored NMT, we get additional training data. Moreover, instead of training distinct models for each dataset, we just Several word embeddings are accessible and free to use however it is not simple to get a character embedding. MT model boosts the performance of other NLP tasks.. We found that a pre-trained encoder of a Machine Translation model boosts the performance of other NLP tasks. Their contextualized word vectors are known as Context Vectors.Broadening this notion, we extract embeddings from character-level Neural Machine Translation model trained with an aligned set[5].

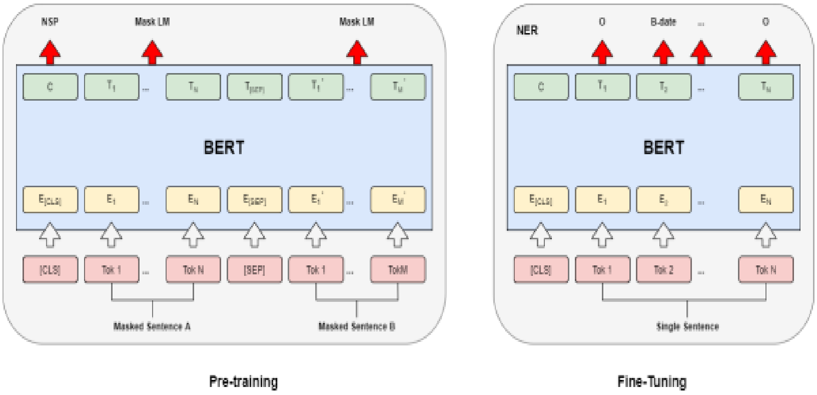


FIGURE 3 Overview of BERT Model

4 | EXPERIMENTAL SETUP

4.1 | BERT

Bidirectional Encoder Representations from Transformers (BERT) is a model trained on bidirectional representations that is built on the transformer architecture. This indicates that BERT can read representations from unlabeled text in both ways. BERT training begins with a pre-training phase in which the model is taught to interpret language and context by performing two unsupervised tasks: Masked language modelling (MLM) and Next Sentence Prediction (NSP) (NSP). BERT may then be fine-tuned to learn a certain activity. For an overview of the pre-training and fine-tuning stages, go here.

4.2 | Dataset

The second round of the ICDAR 2019 competition on post-OCR text correction is presented, as are the many solutions submitted by competitors. OCR has been a study topic for almost 30 years, yet the findings are still poor, particularly for historical texts. This competition compares and evaluates automated systems for repairing (denoising) OCR-ed texts. The current challenge is divided into two parts: 1) error detection and 2) error rectification. The participants were given an original dataset of 22 million OCR-ed symbols together with an aligned ground truth, with 80 percent of the dataset committed to training and 20 percent to assessment. Newspapers, historical written materials, manuscripts, and shopping receipts from ten European languages were gathered from various sources (Bulgarian, Czech, Dutch, English, Finish, French, German, Polish, Spanish and Slovak). Five teams provided findings, and the mistake detection scores ranged from 41 to 95 percent, with 44 percent being the greatest error correction improvement. This competition, which received 34 registrations, demonstrates the community's strong desire to increase OCR output, which is a critical problem in any digitization process involving textual data[6].

4.3 | Convolutional Neural Network

In this model, we employ the pretrained BERT embeddings model bert-base-multilingual-cased in our model. The convolutional layers receive BERT embeddings with four distinct kernel sizes (2, 3, 4, and 5), each with 32 filters. Following the convolutional layers are the maxpool layers. Stride=1 is used by both convolutional and maxpool layers, which has an influence on information sharing inside the n-grams, which are 2-, 3-, 4-, and 5-grams. Finally, the maxpool layers' outputs are concatenated and fed into the linear layer to obtain the final logits for the binary classification. This method, I believe, is analogous to the picture segmentation issue.

4.3.1 | Trainig the Model

Context-based Character Correction (CCC) is our technique, which employs the context-aware pretrained language model BERT. The pretrained multilingual BERT is used in our detection model. Each sub-BERT token's output is then inserted into convolutional layers and fully-connected layers to be categorised. Sub-token model predictions are combined into token-level predictions. If more than one sub-token of a token is projected to be incorrect, the token is incorrect. Our corrective model is a sequence-to-sequence model with an attention mechanism. The encoder is a bidirectional LSTM that shares the character embedding with the decoder. The encoder input consists of characters from erroneous tokens and related context information from the BERT, which has been fine-tuned throughout the detection phase. Character-level adjustments are generated by the decoder.Using beam search, the ultimate rectification of each erroneous token may be discovered.

preparing the inputs for the model: We have 10 langauges in the dataset,Hence we will try to use BERT Multilingual Tokenizer for more than 1 languages.

[tokenizer = BertTokenizer.from$_{pretrained}$*"* $bert - base - multilingual - cased$*"*

The following procedures are used to create sufficient input data for the model :

1. Using BERT's Word piece tokenization, tokenize the words.
2. Converting tokens to identifiers
3. Alignment of sequence lengths to maximum sequence length
4. Making attention masks, where 0 represents padding
5. Tensorization of token ids, labels, and attention masks

5 | CONCLUSION

This research describes an innovative method for improving the quality of digital outputs. Using word embeddings and pre-trained BERT models, our error detector detects a variety of real-world faults. Our correction method, which employs NMT methods on contextual input data and includes certain extra characteristics, seems to have the potential to decrease OCRed mistakes. However, if real-word mistakes are due to incorrect line detection, our approach's performance is restricted.

6 | RESULTS

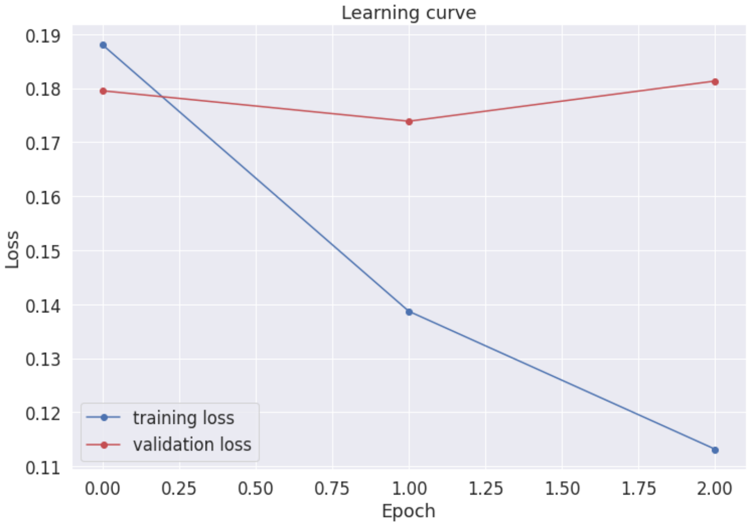


FIGURE 4 Training Loss and Validation Loss

The Ultimate accuracy of the model is 95.6 percent.After epoch 1, the model began to overfit, as predicted. Interestingly, it achieved validation accuracy of more than 95 percent after the zeroth epoch, but training accuracy was substantially lower — approximately 93 percent.

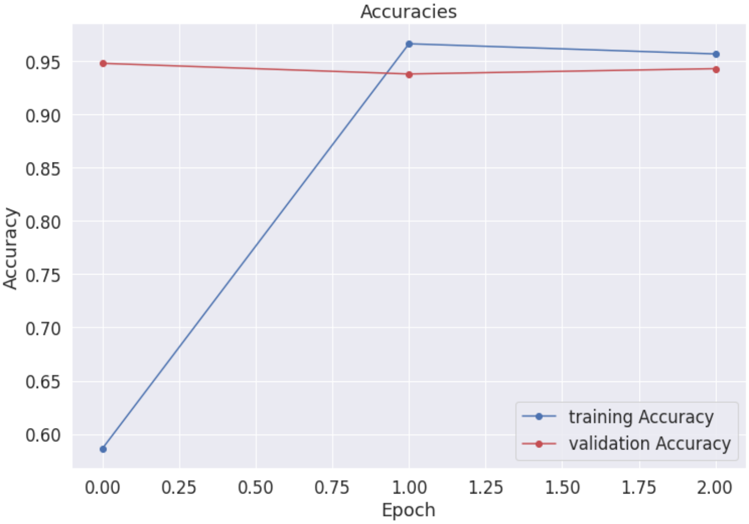


FIGURE 5 Training Accuracy and Validation accuracy

7 | REFERENCES

REFERENCES

[1] Rigaud C, Doucet A, Coustaty M, Moreux JP. ICDAR 2019 competition on post-OCR text correction. In: 2019 international conference on document analysis and recognition (ICDAR) IEEE; 2019. p. 1588–1593.

[2] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805 2018;.

[3] Nguyen TTH, Jatowt A, Nguyen NV, Coustaty M, Doucet A. Neural machine translation with BERT for post-OCR error detection and correction. In: Proceedings of the ACM/IEEE joint conference on digital libraries in 2020; 2020. p. 333–336.

[4] Lund WB, Kennard DJ, Ringger EK. Combining multiple thresholding binarization values to improve OCR output. In: Document Recognition and Retrieval XX, vol. 8658 International Society for Optics and Photonics; 2013. p. 86580R.

[5] McCann B, Bradbury J, Xiong C, Socher R. Learned in translation: Contextualized word vectors. Advances in neural information processing systems 2017;30.

[6] Gao L, Huang Y, Déjean H, Meunier JL, Yan Q, Fang Y, et al. ICDAR 2019 competition on table detection and recognition (cTDaR). In: 2019 International Conference on Document Analysis and Recognition (ICDAR) IEEE; 2019. p. 1510–1515.