Supervised Post-OCR Correction with Neural Machine Translation

PRAT's Final Report

by:
Guillaume THOMAS [1]

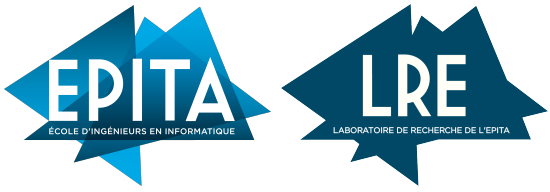
supervised by:
Joseph CHAZALON [1]
Edwin CARLINET [1]

[1] LRDE, EPITA, Kremlin-Bicetre, France

Abstract

While Optical Character Recognition (OCR) systems have enabled the extraction of text from digitalized documents, they still produce noisy outputs when dealing with historical documents. The ANR SODUCO project uses scanned paper directories to analyze social dynamics in Paris during the 17-20th century. However, correcting the extracted text is crucial for downstream Natural Language Processing tasks, but was challenging due to an already low Character-Error-Rate (CER) of around 4%. This report presents a three-stage pipeline that employs state-of-the-art post-OCR correction techniques. The pipeline effectively reduced the Character-Error-Rate (CER) of the raw extracted text by 20%. The pipeline comprises an error detector module based on CamemBERT Transformers, a Neural Machine Translation module that performs the core correction task, and a post-processing module based on length. The notebook containing our experimentation records and pre-trained models is available on this Google Drive [1].

Keywords: OCR; error correction; Neural Machine Translation



[1] https://drive.google.com/drive/folders/1JN8eJDCH5Ws6ALzqVNXOPpok-lxDpqEU?usp=sharing

Contents

1 Introduction

1.1 Context

Optical Character Recognition (OCR) systems enable the extraction of text from digitalized documents, or more generally from images. The field of research around this technology has been active for over the past 30 years and achieves, now, remarkable results on mainstream documents. But those systems still terribly fail on historical documents. Due to their challenging layouts, their storage conditions, or the poor quality of their original printing materials, the extraction pipeline, therefore, produces a very noisy OCR output. This noisy extracted text can be, in the end, strongly diverging from the original text, known as the Ground Truth (GT). This noise can hamper downstream Natural Language Processing (NLP) tasks, which makes difficult the indexation, access, and exploitation of such documents.

The ANR SODUCO project is interested in the social dynamics in the heart of Paris during the 17-20th century. The SODUCO project provides valuable resources for geographic and historical research. Fig. 1 depicts preliminary results of the distribution of professions in Paris during the year 1850. To extract such information, the project relies on scanned paper directories from which the text has been extracted. Even with specific OCR well-suited for historical documents [2]([13], [12], [14]), the text remains quite noisy. An OCR post-processing is then required to correct this noise. Post-OCR approaches are also well-suited in other scenarios in which OCRization is costly and cannot be performed again or if the digitization pipeline is reserved to process newly arrived documents.

1.2 Formalized problem

The OCR-noise correction process can be characterized as a Machine Translation task, wherein the aim is to translate a given text from one language to another. In the context of post-OCR correction, the Machine Translation model is trained to "translate" an erroneous language version to a corrected one. Unlike traditional language translation settings, the input and output for this process rely on the same character and sub-word vocabularies, instead of having two separate vocabularies.

[2] `https://github.com/DCGM/pero-ocr`

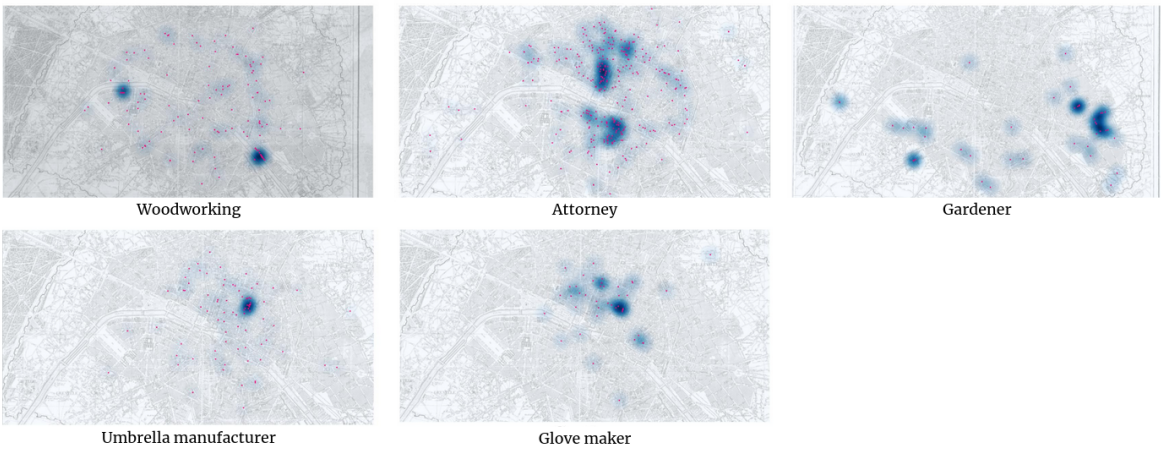


Figure 1: Preliminary results of the SODUCO project, extracted from Didot-Bottin, 1850

1.3 Metrics

Our study was based on state-of-the-art, simple-to-use and simple-to-interpret metrics, namely the micro-averaged and the macro-averaged Character-Error-Rate.

Character-Error-Rate (CER). The Character-Error-Rate represents the percentage of characters that were erroneously predicted. Lower CER values indicate higher system performance. While CER traditionally requires character alignment between the predicted and ground-truth sequences, this limitation can be circumvented by using its definition in equation 1, based on the Levenshtein distance.

$$CER = \frac{substitution + deletion + insertion}{substitution + deletion + correct\ charaters} \tag{1}$$

The Character Error Rate (CER) still presents some downsides. For instance, it is sensitive to variations in text length, as longer texts are more prone to errors. Additionally, the CER may exhibit bias towards formatting and punctuation errors. This is because letter substitutions are given the same weight as punctuation substitutions or deleted spaces, despite the former being considerably more meaningful errors. The CER provides the advantage of supplying detailed character-level information (in contrast to the Word Error Rate) and enjoying broad utilization in the state-of-the-art, facilitating comparisons with current standards.

Macro and micro-average. This study utilized both micro-average and macro-average methods for aggregating the Character Error Rate (CER) of all samples into a singular metric. The macro-average method first computes the CER on each sample and subsequently calculates an unweighted mean of the results. Conversely, the micro-average method directly calculates the CER over the concatenated samples. As a result, these two metrics present distinct hypotheses. The micro-average method considers the varying lengths of the samples while the macro-average method assumes equal length for all samples. This difference is significant and will have implications for future analyses.

To average the CER of all samples into a single metric, we used the micro and macro average. While the macro-average first compute the CER on every samples and performs an unweighted mean over those, the micro-average is directly computing the CER over the whole concanated samples. Those two metrics, therefore, present different hypthesis. The micro-average performs a weigthed mean depending on samples' length, and the macro is performing it unweightedly, i.e. by supposing an equivalent length for all samples. This difference will prove itself to be important in later stages.

Summed Levenshtein distances. The ICDAR competition in 2017 ([5]) and 2019 ([20]) proposed the metric of Summed Levenshtein distances for the semi-supervised error correction task. The competition

provided a publicly available dataset and an evaluation tool[3], which aimed to establish a new standard and enable meaningful comparisons in this domain. Nevertheless, the application of Summed Levenshtein distances in a fully automatic pipeline would be equivalent to an unnormalized CER, which lacks relevance and presents significant redundancy with default CER. Therefore, we opted not to utilize this metric in our pipeline.

1.4 Dataset

Table 1: Qualitative examples of errors

Extracted text	Ground Truth text
Castries\n(Cte de\n" Varennes, 22.	Castries (Cte de), Varennes, 22.
Legrand ainé, baleines, r. des Fontaines, 15.	Legrand aîné, baleines, r. des Fontaines, 15.
Berthier, Pl. Vendome, III.	Berthier, Pl. Vendome, 111.
Chomet, R. du Fb. 3. Honoré, 39.— Ch. Elysées.	Chomet, R. du Fb. S. Honoré, 39.— Ch. Elysées.

The present study utilized a dataset consisting of 8,392 samples, which were manually annotated and similar to the ones presented in Table 1. The samples were derived from an extraction pipeline. First, the structure of each page of each directory is extracted, including the column separators and text areas specific to each individual. Subsequently, an Optical Character Recognition (OCR) system extract the text from each of the identified regions. We employed an OCR system specifically designed for processing old documents, Pero OCR.

Figure 2: Typical layouts of directories

It is pertinent to note that the structure of the dataset was manually corrected prior to OCR extraction. Thus, the dataset solely includes OCR errors and not any structure errors. This limited scope simplifies the solution pipeline, as a more intricate pipeline would be necessary to separate a single sample into two different ones that the structure extraction would have grouped together.

Furthermore, the original dataset already exhibits a relatively low Character Error Rate (CER), a micro-CER of 3.76%, and a macro-CER of 4.17%. Each sample features only a small number of substitution

[3] `https://git.univ-lr.fr/gchiro01/icdar2017`

errors, such as the replacement of "I" with "1", as shown in the third row of Table 1. This characteristic will prove to be significant in later stages.

1.5 Contribution

This scientific report presents a three-stage pipeline that employs state-of-the-art post-OCR correction techniques. The pipeline effectively reduced the Character-Error-Rate (CER) of the raw extracted text by 20%. The pipeline comprises an error detector module based on CamemBERT Transformers, a Neural Machine Translation module that performs the core correction task, and a post-processing module based on length. The notebook containing our experimentation records and pre-trained models is available on this Google Drive [4].

2 Related work

[23] proposed three approaches to address OCR errors, namely modifying input images, altering OCR systems, and post-processing output text. Among these options, many recent approaches have focused on post-processing because it does not involve re-OCRing. Traditional methods for post-correction involved statistical language modeling techniques, in combination of string distance metrics and n-gram ([22], [15]). However, more recent approaches use general string-to-string translation methods that achieve significantly better results ([10]). [2] trained a statistical machine translation system to correct OCR errors in historical French texts, which outperformed language model-based techniques.

Traditional statistical machine translation techniques used to divide a sentence into multiple parts and provide a translation for each one, which resulted in translations that lacked fluency and were often out of order due to different grammar rules. Neural Machine Translation (NMT) has emerged as a more advanced approach ([20], [18], [3]) that uses an encoder-decoder architecture with an attention mechanism ([4]) to achieve better text correction results ([7]), thanks to a focus on the whole source sentence.

When implementing encoders and decoders for sequential data of varying size like sentences in OCR post-correction using neural network architectures, Recurrent Neural Networks (RNNs) are the most suitable option. RNNs come in different architectural flavors, with the two most commonly used RNN cells being Long-Short-Term Memory (LSTM)([11]) and Gated Recurrent Unit (GRU)([6]) cells.

The ICDAR competition on post-OCR text correction provides an overview of the recent state-of-the-art in OCR post-correction. The competition includes two tasks, error detection, and error correction, with evaluation sets for English and French languages. In the 2017 edition ([5]), Weighted Finite-State Transducers (WFST) obtained the best results in the error detection task, while a SMT/NMT system achieved the highest reduction of errors in the error correction task. In the 2019 edition ([20]), an NMT correction system supported by a detection system based on Bidirectional Encoder Representations from Transformers (BERT) ([8]) achieved the best performance. The ICDAR competition dataset and evaluation pipeline have been made publicly available for meaningful comparisons in the field [5].

3 Solution Pipeline

3.1 Correction model with Neural Machine Translation

Our OCR post-correction pipeline relies on a sequence-to-sequence translation model, which is implemented using an Encoder-Decoder architecture with Long Short-Term Memory (LSTM) at the character-level. To implement this architecture, we utilized the Open Neural Machine Translation (OpenNMT) toolkit, which is widely recognized and often used in its default settings which facilitate comparison.

[4] `https://drive.google.com/drive/folders/1JN8eJDCH5Ws6ALzqVNXOPpok-lxDpqEU?usp=sharing`

[5] `https://git.univ-lr.fr/gchiro01/icdar2017`

We opted to process the sample at the character-level for several reasons. This approach allows for the correction of unseen words during training, which is impossible when using token-level processing. Additionally, it requires less data, as character tokens outnumber SentencePiece tokens, which should facilitate learning the relationships between tokens. Lastly, processing at the character-level matches more OCR errors and still greatly includes context. This is why the current state-of-the-art is now exclusively focused on character-level one ([20], [5], [19], [3], [17], [9]). Moreover, recent benchmark ([17]) on NMT models' performance at both word and character levels showed clear conclusion regarding the character-level approaches.

However, this naive approach results in double the Character Error Rate (CER), with macro-CER increasing from 4.17% to 7.99%.

3.2 Length post-processing

The current results of the correction process are inconclusive. However, we observed a significant divergence between the micro and macro averages. Through an analysis of the correlation between the length of the samples and their corresponding corrections, we concluded that a length post-processing step was necessary. This post-processing step is performed in two stages: the detection of length aberrations and the recovery of the original uncorrected sample. The implementation of this post-processing step resulted in a significant improvement over the baseline performance, with a relative increase of 10%.

Firstly, we observed a significant difference between micro and macro averages (14.31% vs 7.99%). As discussed before, the micro-average method considers the varying lengths of samples, while the macro-average method assumes equal length for all samples. The significantly higher micro-average highlights a correlation between longer samples and higher CER.

Secondly, upon closer examination of the data presented in Fig. 3, a comparison can be made between the lengths of the samples and their corresponding corrections. It is expected that there will be a high correlation between these two lengths, as each sample should only undergo a few substitutions. While the majority of samples adhere to this rule, there are certain cases of "text summarization" and "text stuttering," in which the corrected sample is significantly shorter or longer, respectively. These phenomena are qualitatively illustrated in Table 2.

Table 2: Qualitative examples of text summarization and text stuttering

Text stuttering
Text summarization

Text stuttering has been identified as the most detrimental factor for micro-CER. This can be attributed to its larger weight in micro-average due to its length and higher CER, which can sometimes exceed 100%. As a result, it leads to significantly higher micro-CER. Text summarization is also problematic, with a significant CER. Correcting these aberrations is crucial for improving the performance of our model.

The correction process is performed in two steps. Firstly, a detection method based on the length is employed. Then, a length regularization operation is performed. Two different length regularization operations were tested: the "Recovery" strategy, which returns the original sample, and the "Cut" strategy, which shortens the corrected sample to the length of the original sample. The results showed that the "Recovery" strategy was the most effective, achieving a macro-average CER of 3.72%, while the "Cut" strategy had a CER of 7.30%.

Text stuttering and text summarization are detected using a straightforward threshold method. The threshold value was selected to minimize the CER over the training set. Text samples that are 40% longer or 10% shorter are then subjected to the "recovery" strategy for post-correction. This post-processing method has resulted in the first significant improvement over our baseline performance. The macro-average CER decreased from 4.17% to 3.72%, indicating a relative improvement of 10.8%.

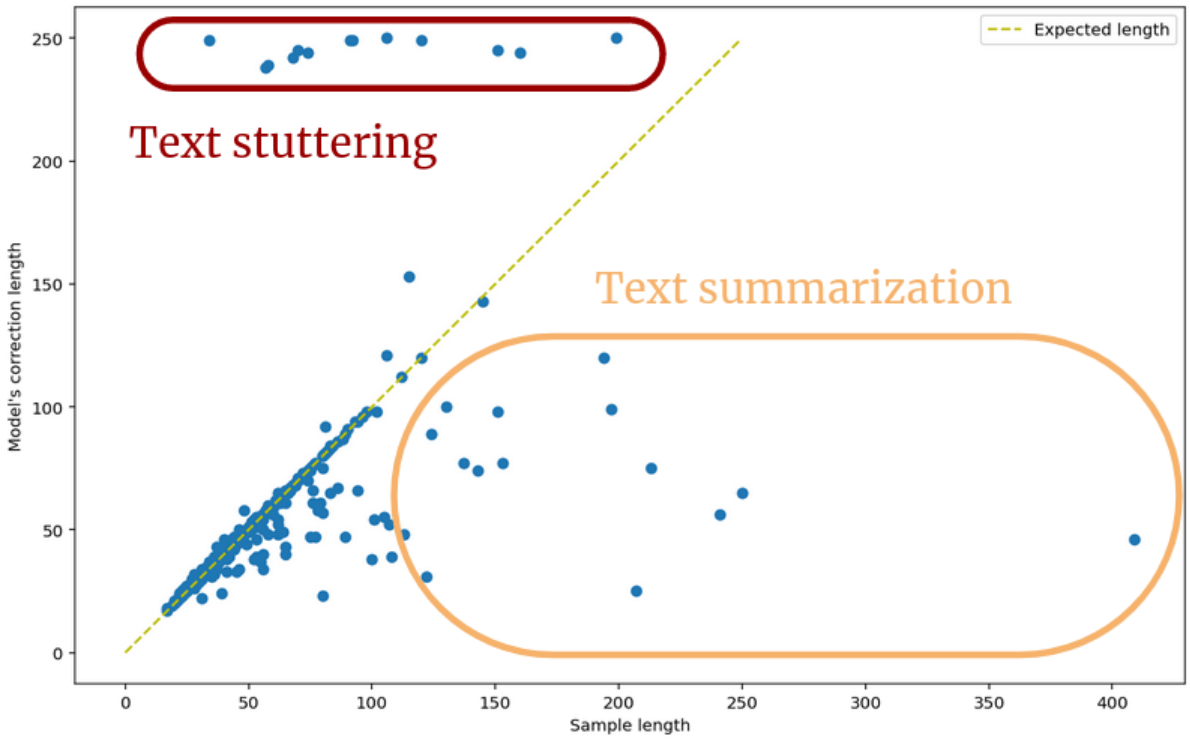


Figure 3: Comparison of sample's length and their correction's length

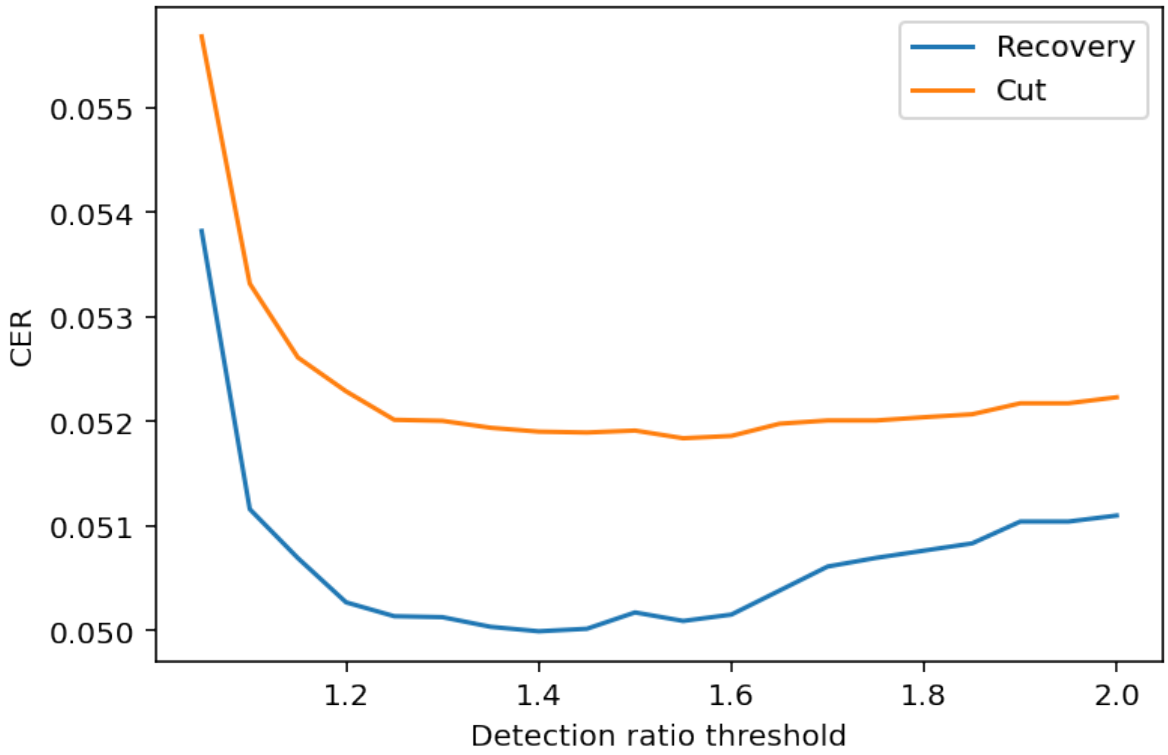


Figure 4: Benchmark of "Cut" and "Recovery" strategies

3.3 Error detector

The study revealed that the dataset consisted of 50% of samples that were error-free. Despite this, the correction model applied edits to all samples, which produced false positive corrections. To mitigate this issue, a detection model is needed to filter out incorrect corrections before they are made. Implementing this additional stage resulted in a significant reduction of the macro-CER to 3.33%, indicating a 20% relative improvement compared to the baseline.

Robin et al. [21] also encountered this issue when they used a dataset with low CER (around 1.1%). To improve their correction model's performance, they proposed a pre-processing detection step. This step aimed to identify and filter out incorrect corrections before they were made. By doing so, the authors achieved several advantages. The correction model is trained solely on samples that contain errors. Furthermore, it artificially increases the number of errors that the correction model sees during training, which can improve its performance. Additionally, this method prevents the bias of adding correction to every sample, effectively preventing false positive corrections.

In contrast to Robin et al. [21], who utilized a bidirectional LSTM as their detector model, our approach employs a more recent architecture. Inspired by the winner of the ICDAR 2019 competition ([20]) and [19], we based our implementation on a BERT-like model. Specifically, we utilized CamemBERT [16], a multi-layer bidirectional transformer pre-trained language model designed for French. We conducted error detection at the sentence level, utilizing the CamembertForSequenceClassification module, which was fine-tuned for this specific purpose [6].

3.4 Experimentation with Named Entities as Meta Data

In this study, we aimed to improve the performance of our correction model by utilizing a priori information about each token. Specifically, our pipeline incorporate a Named Entity Recognition (NER) system that tag entities such as addresses, professions, and patronymes on each sample. We attempted to train a single Neural Machine Translation (NMT) model using the regrouped character tokens under each named entity, hoping that it would learn different correction patterns for each named entity. However, our results were inconclusive, with a 4% increase in errors relative to the previous correction model. The correction model specialized for Named Entities had a macro-CER of 8.31%, against 7.99% for the previous correction model alone.

The NER module in our pipeline was found to be robust to OCR-noise [1], making it a valuable source of robust and pertinent annotation for our correction model. Previous studies [3] have explored the inclusion of context information at the character level, such as text type and time span when the text was written. In our study, we leveraged the named entities as factors to train the NMT model to learn different correction patterns for each named entity, as the vocabulary associated with professions and names is quite distinct.

However, our attempts at training a single NMT model with a priori named entity information were unsuccessful. The reason for this failure remains unclear, but it is possible that no correction distinction between named entities is necessary. Another possible factor is the reduced size of the training set by a factor of four, which may have limited the ability of the NMT model to learn. Specifically, the reduction of the training set size was due to the need to regroup tokens under the four named entities.

3.5 Full stage pipeline

The figure 5 presents a summary of the complete three-stage pipeline. The detection model is responsible for distinguishing the extracted text samples that are accurate from those that contain errors. The samples that are identified as correct remain unaltered, while the incorrect samples are subjected to correction by the Neural Machine Translation-based correction model. However, this model may produce erroneous

[6] `https://huggingface.co/docs/transformers/model_doc/camembert#transformers.CamembertForSequenceClassification`

corrections for a few rare samples. To remedy such aberrations, a post-processing stage is employed, which uses length-based correction. Only the corrected samples whose length is highly correlated with the original length are retained, while the other corrected samples are discarded, thereby returning the original sample.

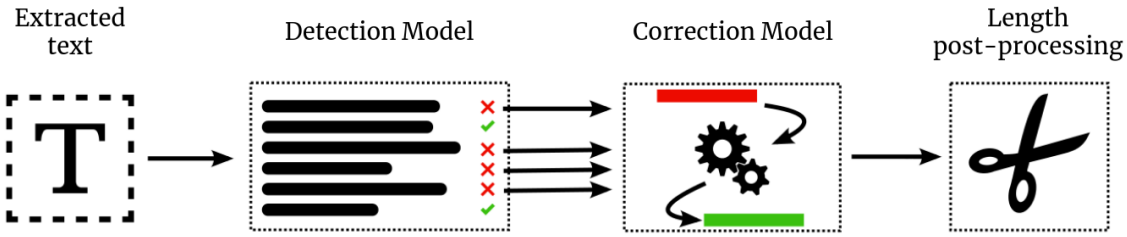


Figure 5: Full pipeline

In order to test the relevance of each component of our pipeline, we performed an ablation study. It involves systematically removing one or more elements of the model and analyzing the effect of the removal on the overall performance of the model. The results are presented in table 3.

Table 3: Performance associated to the ablation study

	Micro-CER	Macro-CER
Baseline	3.76%	4.17%
Correction model	14.31%	7.99%
Correction model + Length post-processing	3.98%	3.72%
Detection and Correction model	7.87%	5.68%
Detection and Correction model + Length post-processing	3.51%	3.33%

4 Conclusion

This report presented a three-stage pipeline that employs state-of-the-art post-OCR correction techniques. The pipeline comprises an error detector module based on CamemBERT Transformers, a character-level Neural Machine Translation module that performs the core correction task, and a post-processing module based on length. The notebook containing our experimentation records and pre-trained models is available on this Google Drive [7].

The evoked pipeline was found to significantly reduce the Character-Error-Rate (CER) of the raw extracted text by 20%. However, it should be noted that the correction is not uniform across part of a sample. Specifically, the table 4 shows that text containing letters (such as person, profession or address) had the lowest macro-CER, ranging from 1.40% to 2.40%. In contrast, the cardinal, which is composed of digits, had a significantly higher macro-CER, averaging around 4.40%. This can be explained by the greater difficulty of correcting numbers based on context, such as discerning between "11" and "17", compared to the correction of letters, such as recognizing that "deg" is not a word while "dog" is.

Qualitatively speaking, the predominant correction focuses on the punctuation level. The first entry of table 5 greatly illustrates these corrections. This correction method successfully standardizes input entries and has proven to be highly effective; in fact, we have found that certain samples within the ground truth lack proper punctuation as a result of this normalization process. Although some character errors do exist, they are relatively infrequent and are further exemplified below.

[7] https://drive.google.com/drive/folders/1JN8eJDCH5Ws6ALzqVNXOPpok-lxDpqEU?usp=sharing

Table 4: Results summary per Named Entities

Method	*Raw text*		*Corrected text*	
Metric	*Micro-CER*	Macro-CER	Micro-CER	Macro-CER
Global	3.76%	4.17%	3.51%	3.33%
Person	2.84%	2.53%	2.02%	1.97%
Profession	2.52%	1.55%	2.53%	1.41%
Address	3.47%	2.42%	2.40%	2.39%
Cardinal	7.23%	7.69%	4.33%	4.45%

Table 5: Some qualitative results

Extracted text	Corrected text
Castries\n(Cte de\n" Varennes, 22.	Castries (Cte de), Varennes, 22.
Legrand ainé, baleines, r. des Fontaines, 15.	Legrand aîné, baleines, r. des Fontaines, 15.
Berthier, Pl. Vendome, III.	Berthier, Pl. Vendome, 111.
Chomet, R. du Fb. 3. Honoré, 39.— Ch. Elysées.	Chomet, R. du Fb. S. Honoré, 39.— Ch. Elysées.

In conclusion, there are still several areas that can be explored to enhance the performance of our pipeline. The following is a non-exhaustive list of these areas: Firstly, it would be beneficial to experiment with different post-processing methods, such as post-processing based on the number of edits instead of the length of the text. Secondly, a detection model at the character-level should be investigated to improve recall, which is currently at 67%, and could provide additional guidance to the correction model. Thirdly, the unsupervised huge dataset could be exploited to improve the correction model by using it as an unsupervised pre-training method. Fourthly, more modern architectures, such as Transformers, could be explored in the correction model instead of LSTM. Finally, different tokenization techniques, like SentencePiece, should be investigated in the correction model.

References

[1] N. Abadie, E. Carlinet, J. Chazalon, and B. Duménieu. A benchmark of named entity recognition approaches in historical documents application to 19 th century french directories. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13237 LNCS:445–460, 2022.

[2] Haithem Afli, Loïc Barrault, and Holger Schwenk. Ocr error correction using statistical machine translation. *International Journal of Computational Linguistics and Applications*, 7:175–191, 2016.

[3] Chantal Amrhein and Simon; https://orcid.org/0000 0003-1365-0662 Clematide. Supervised ocr error detection and correction using statistical and neural machine translation methods. *Amrhein, Chantal; Clematide, Simon (2018). Supervised OCR Error Detection and Correction Using Statistical and Neural Machine Translation Methods. Journal for Language Technology and Computational Linguistics (JLCL), 33(1):49-76.*, 33:49–76, 2018.

[4] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 9 2014.

[5] Guillaume Chiron, Antoine Doucet, Mickael Coustaty, and Jean Philippe Moreux. Icdar2017 competition on post-ocr text correction. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 1:1423–1428, 7 2017.

[6] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *Proceedings of SSST 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, 9 2014.

[7] Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. Neural network translation models for grammatical error correction.

[8] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:4171–4186, 10 2018.

[9] Rui Dong and David A. Smith. Multi-input attention for unsupervised ocr correction. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:2363–2372, 2018.

[10] Steffen Eger, Tim Vor Der Brück, and Alexander Mehler. The prague bulletin of mathematical linguistics a comparison of four character-level string-to-string translation models for (ocr) spelling error correction. 2016.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 11 1997.

[12] Martin Kišš, Karel Beneš, and Michal Hradiš. At-st: Self-training adaptation strategy for ocr in domains with limited transcriptions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12824 LNCS:463–477, 2021.

[13] Oldřich Kodym and Michal Hradiš. Page layout analysis system for unconstrained historic documents. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12822 LNCS:492–506, 2021.

[14] Jan Kohút and Michal Hradiš. Ts-net: Ocr trained to switch between text transcription styles. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12824 LNCS:478–493, 2021.

[15] Okan Kolak and Philip Resnik. Ocr error correction using a noisy channel model. 2002.

[16] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. Camembert: a tasty french language model. pages 7203–7219, 11 2019.

[17] Kareem Mokhtar, Syed Saqib Bukhari, and Andreas Dengel. Ocr error correction: State-of-the-art vs an nmt-based approach. *Proceedings - 13th IAPR International Workshop on Document Analysis Systems, DAS 2018*, pages 429–434, 6 2018.

[18] Thi Tuyet Hai Nguyen, Adam Jatowt, Mickael Coustaty, and Antoine Doucet. Survey of post-ocr processing approaches. *ACM Computing Surveys (CSUR)*, 54, 7 2021.

[19] Thi Tuyet Hai Nguyen, Adam Jatowt, Nhu Van Nguyen, Mickael Coustaty, and Antoine Doucet. Neural machine translation with bert for post-ocr error detection and correction. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, pages 333–336, 8 2020.

[20] Christophe Rigaud, Antoine Doucet, Mickael Coustaty, and Jean Philippe Moreux. Icdar 2019 competition on post-ocr text correction. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 1588–1593, 9 2019.

[21] Schaefer Robin and Clemens Neudecker. A two-step approach for automatic ocr post-correction. pages 52–57, 2020.

[22] Xiang Tong and David A Evans. A statistical approach to automatic ocr error correction in context. 1996.

[23] Martin Volk, Lenz Furrer, and Rico Sennrich. Strategies for reducing and correcting ocr errors. *Language Technology for Cultural Heritage*, pages 3–22, 2011.