OCR cleaning of scientific texts with LLMs

Gábor Madarász[1], Noémi Ligeti-Nagy[1][0000−0003−0851−7621], András Holl[2][0000−0002−6873−3425], and Tamás Váradi[1][0000−0001−5765−3908]

[1] HUN-REN Hungarian Research Centre for Linguistics, Budapest, Hungary
{surname.firstname}@nytud.hun-ren.hu
[2] Library and Information Centre, Hungarian Academy of Sciences, Budapest, Hungary
holl.andras@konyvtar.mta.hu

Abstract. Correcting Optical Character Recognition (OCR) errors is a major challenge in preprocessing datasets consisting of legacy PDF files. In this study, we develop Large Language Models specially finetuned to correct OCR errors. We experimented with the mT5 model (both the mT5-small and mT5-large configurations), a Text-to-Text Transfer Transformer-based machine translation model, for the post-correction of texts with OCR errors. We compiled a parallel corpus consisting of text corrupted with OCR errors as well as corresponding clean data. Our findings suggest that the mT5 model can be successfully applied to OCR error correction with improving accuracy. The results affirm the mT5 model as an effective tool for OCR post-correction, with prospects for achieving greater efficiency in future research.

Keywords: OCR errors · Large Language Models · mT5 model · natural scientific language processing.

1 Introduction

This paper reports on a collaborative project between the HUN-REN Hungarian Research Centre for Linguistics (HUN-REN NYTK) and the Library and Information Centre of the Hungarian Academy of Sciences (MTA KIK)designed to make the content of the REAL Repository more easily accessible to researchers and more easy to curate and enhance for MTA KIK. Prior to embarking on the data-mining of the texts in the Repository, the files have to be converted to machine readable raw text format. The paper will focus on techniques to clean the texts of OCR errors, which is a major challenge in this preprocessing phase. Our strategy is to compile parallel corpora consisting of sentences with OCR errors and their correct counterparts, which are used as training date to finetune a large language model so as to enable it to correct badly OCR'ed texts. The Structure of the paper is as follows: Section 2 describes the context and the motivation for the work, Section 3 reviews related work in OCR cleaning, Section 4 elaborates the various datasets used for the training corpus, Section 5 contains a brief description of the training method, section 6 enumerates and discusses the results and finally, the paper ends with some Conclusions.

2 Motivation

The Library of the Hungarian Academy of Sciences was established in 1826, and since then it has been serving the members of the Academy and the whole Hungarian research community. Besides its main collection, the library has a special collection of manuscripts and rare books, and an Oriental Collection as well. The digital collections – in the form of an open access repository – were created in 2008. This repository – named REAL – has diverse holdings, mirroring the printed collection of the library. Its content is partly based on an extensive digitisation project and it contains born-digital materials too (e.g. modern journals within the scope of our library). The third source of material is the OA mandate of the Academy – researchers supported by the Academy are mandated to reposit their output in REAL. The diversity of input channels results in a mixed document content – scanned and born digital, publishers' PDFs and accepted manuscripts (with an assortment of handwritten documents and images to top it up).

The original goal of the repository was to supply digital documents for the researchers. We store PDF documents (most of which have a text layer) and the inclusion criterion was that they are suitable for the human user. Each document is checked by a librarian, so some basic document and metadata quality can be guaranteed. On the other hand, we are aware of the problems of OCR (or occasionally, the lack of it), the errors and gaps in the meta-data.

The question of language information for the documents is such a problem. Human users can obviously perceive whether a document is written in a language that is accessible for them, but we cannot filter search results for language. The lack of document language information was an early setback for our project.

The REAL Repository contains more than 250 thousand documents, about a half of which, amounting to one billion words, is suitable for the project.

The Library's most fundamental goal with this project is to enhance meta-data (e.g. provide detailed language information). We would also like to improve the quality of the text layer, correcting errors in the OCR, and provide clean text layers for search and text mining.

Furthermore, we would like to be able to recognise named entities in the text. One specific task we would like to accomplish is finding references to other publications. Similarly, references to grants, large research facilities and software are also of interest to MTA KIK. (The library operates the national bibliographic database, a CRIS-like system.)

In summary, we would like to improve the data and metadata quality, text-mine information for scientometric (and other) purposes, and improve the efficiency of search.

3 Related work

There is a growing interest in utilizing neural technologies for post-OCR text correction. One of the few studies specifically addressing the correction of Hungarian

texts using neural technologies is by Laki et al. [3], who explored four distinct correction experiments: machine translation with the Marian neural machine translation (NMT) system, fine-tuning a Hungarian BART model for machine translation, Context-based Character Correction (CCC) combined with machine translation using the Marian NMT system, and CCC detection with fine-tuning of Hungarian BART for machine translation.

Another notable work in this area includes research on Sanskrit texts by Maheshwari et al. [4], who reported a significant improvement in Character Error Rate (CER) using mT5 (+14.1%) and ByT5 (+23.4%) models. Piotrowski [5] focused on the application of pre-trained language models for OCR post-correction, achieving a 4.3% word error rate improvement by fine-tuning mT5 and plT5 models.

Alternative approaches to OCR correction have also been explored. Rigaud et al. [7] introduced the ICDAR2019 winning OCR correction method CCC, which combines a convolutional network for detection with a correction mechanism utilizing a BERT model and a bidirectional LSTM (Long Short Term Memory) model with an attention mechanism. Schaefer and Neudecker [8] proposed a two-step approach that includes OCR error detection with a bidirectional LSTM and subsequent error correction with a sequence-to-sequence translation model. Furthermore, Gupta et al. [2] implemented an unsupervised multi-view post-OCR error correction technique employing GPT, GPT2, and GPT2XL autoregressive models, benchmarked against a 3-gram model trained on Wikipedia. Lastly, Amrhein [1] addressed OCR error correction using a character-based NMT approach, showcasing the versatility of neural methods in enhancing OCR accuracy across various languages and scripts.

4 The training data

When creating the training data, we ensured that the model should be able to identify when to leave the text unchanged by including both error-free and OCR erroneous sentences, with a distribution of 33.6% error-free to 66.4% erroneous data. The dataset comprises 1,355,963 sentence pairs, encompassing a total of 51,658,231 words, with an average sentence length of approximately 19 words. The average Character Error Rate (CER) across the entire training dataset is 12.354%, and the Word Error Rate (WER) is 11.739%, when measured against the reference data (error-free sentences).

The training data was compiled from several sources, detailed below.

4.1 The "JIM corpus"

The construction of a parallel training corpus for OCR correction involved selecting a substantial volume of text available in both electronic (error-free) and OCR-processed (erroneous) versions. This selection was manually or semi-automatically annotated, and then corrected by annotators, leading to the creation of the "JIM" corpus. The process used the complete works of **J**ókai and

Mikszáth, two famous Hungarian writers, chosen for their availability in electronically published formats by the publisher, facilitating a comparison between non-OCR and OCR-processed texts.

The initial challenge was the consolidation of all works by Jókai and Mikszáth into individual files, as each author's works were originally contained in a single file. By following the order of works listed on the https://szaktars.hu website and using a script based on the titles, the works were successfully separated into individual files. This meticulous organization was essential for matching the texts with their corresponding OCR-processed versions, which included additional elements like title pages and indexes not present in the digital editions.

Following the separation of works into individual files, the next step was the construction of a parallel corpus. This involved identifying the OCR-processed counterparts of each work and mapping them at file level, a task complicated by the digital edition containing only the text body, whereas the OCR versions included the complete books. Furthermore, inconsistencies in the availability and order of texts between the OCR versions and digital editions necessitated manual file matching. The subsequent segmentation of these works into smaller units for parallel processing was achieved through sentence-level segmentation and a novel rolling window segmentation method, addressing various challenges such as text normalization and word separation issues.

The parallel corpus underwent semi-automatic annotation to identify and categorize OCR errors, coherence issues, and punctuation differences arising from variations between editions. This process involved listing and prioritizing differences between the OCR and silver texts, ensuring that only OCR-related errors were considered during model evaluation.

Finally, the parallel corpus also underwent further manual correction by four annotators to address discrepancies caused by different editions, using both the error-containing OCR output and the error-free digital text for guidance. Corrections were made with reference to the original PDFs to align the digital text with the version from which the OCR was generated, without strictly adhering to the PDF layout or typographical errors present in the original. Adjustments included adding missing sentences from the OCR to the digital text, ignoring word breaks caused by hyphenation in the OCR that matched the PDF, and not incorporating hyphenation or page numbers from the PDF into the corrected text. The principle behind these corrections was to focus on discrepancies between the OCR text and the corrected version, aiming for textual integrity rather than slavish adherence to the original PDF formatting, especially regarding spacing around punctuation and treatment of hyphenation and page numbers.

The final version of the JIM corpus contained 646 478 sentences (OCR-ed and digital each).

4.2 The datamaker pipeline

A parallel corpus generated from the REAL repository materials consists of parallel sentences extracted using the pdftotext utility (version 0.86.1) from original texts produced during scanning and OCR-ed texts using Tesseract 5.0. A fully

automatic pipeline processes the texts, arranging the raw texts into a training data format suitable for T5-based models.

T5 (Text-to-Text Transfer Transformer [6]) is an encoder-decoder model that converts all NLP problems into a text-to-text format. It is trained using teacher forcing, which means that for training, we always need an input sequence and a corresponding target sequence. The input sequence is fed to the model using `input_ids`. The target sequence is shifted to the right by being prepended with a start-sequence token and is fed to the decoder using `decoder_input_ids`. In teacher-forcing style, the target sequence is then appended with the EOS (end-of-sequence) token and corresponds to the labels. However, it's important to note that the PAD token is not used as the start-sequence token. Instead, a separate token (typically designated as a special token like <s> or similar) is used to signify the start of a sequence. The PAD token is used to fill out sequences for batching purposes so that all sequences in a batch have the same length.

Phase 1: Rule-Based Preprocessing

- Remove sentence separation using the Hungarian tokenizer Quntoken[3].
- Remove newline characters.
- Tokenize sentences using huSpaCy and apply some filtering criteria:
  1. Filter sentences based on the number of tokens ($8 < \text{token\_count} \leq 500$). This step is based on the observation that sentences shorten than 8 tokens usually contain only little information; on the other hand, the maximum number of tokens is specified as 500 because of the `max_token` value of the model (512).
  2. Filter Languages other than Hungarian (only keep sentences detected as Hungarian).
  3. Exclude sentences containing only digits.
  4. Filter sentences with numbers + special character to letter ratio exceeding 0.2.
  5. Exclude sentences with words longer than 30 characters.
  6. Replace commas within quotes in each sentence – One of the most common OCR errors in Hungarian is that the quotation mark characters („) are recognised by the OCR software as double commas, so we replace these by a rule-based approach where necessary.
  7. Remove spaces before punctuation.

Phase 2: Sentence Pairing Based on Similarity

We match the original and Tesseract sentences based on similarity calculated using the NYTK/sentence-transformers-experimental-hubert-hungarian Sentence Transformer model, the huSpacy hu_core_news_lg model, and the Python difflib SequenceMatcher algorithm. Sentences are classified as error-free if all three similarities equal 1.0. During pairing, only sentences with a specified threshold similarity value are included in the database, avoiding the inclusion of sentence pairs with similar meanings but different syntax. This method increased the database by 451,820 sentence pairs.

[3] https://github.com/nytud/quntoken)

4.3 Synthetic data

In the process of creating the Gold Standard Corpus, Laki et al. [3] conducted a comprehensive error analysis, identifying 8,593 distinct OCR error types with the assistance of human annotators. This analysis provided insight into the frequency of various OCR errors. Using these findings, we developed a tool capable of generating synthetic corpora of practically unlimited size. This tool simulates OCR errors by replacing random characters with corresponding OCR erroneous pairs and by inserting or deleting characters, while throughout keeping to the observed frequency of OCR errors in the error-free texts of scanned newspapers. As a result, our training database was augmented with an additional 257,665 lines, significantly improving the diversity and representativeness of our training data.

Figure 1 shows the proportion of the above data sources in the training dataset.

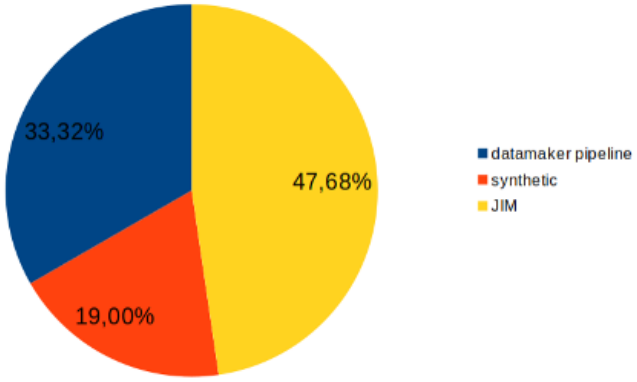

Fig. 1. Proportion of the data in the training dataset

4.4 The gold standard corpus

In parallel with the training and testing of the models we also started the development of a gold standard corpus, which involved a thorough annotation process to ensure that the textual data closely mirrored the original PDFs from which it was derived. This process was rooted in texts extracted from random sections of files from the books of two major publishing companies, covering a wide range of domains. The final corpus contains 100,000 lines, with each line representing a paragraph from the original text, potentially comprising multiple sentences or occasionally being an empty line for structural purposes.

In the first phase of annotation, annotators were tasked with comparing the content of a given PDF to its text (.txt) version created from the PDF, adjusting

the text to match the PDF exactly. This step involved two annotators reviewing and annotating each file independently to ensure thoroughness. Subsequently, their outputs were merged to form a single, finalized version of the text. This rigorous process was guided by key principles designed to retain the original formatting and errors present in the PDFs, excluding page numbers and ensuring correct text structuring, such as maintaining paragraph integrity, differentiating between document sections with double line breaks, and accurately representing dialogue, content lists, images, tables, and footnotes as per specific guidelines.

The annotation principles emphasized the importance of character-level fidelity to the PDF content, even preserving typographical errors. Modifications excluded page numbers and end-of-line hyphenations unless they contributed to the meaning or structure. Text structuring guidelines were strictly followed, including spacing around titles and paragraphs, separation of documents within a volume, and the handling of dialogue units, content lists, images, and tables with appropriate placeholders. Special characters were replaced with their Unicode equivalents, and footnotes were tagged accurately, ensuring that they reflected their placement in the PDF. This detailed approach resulted in a corpus that, while preserving the essence and layout of the original documents, facilitated easier handling and processing for research purposes.

5 The training method

The training data was randomly partitioned into two sets: 90% for training and 10% for testing. We fine-tuned the `google/mt5-large` model using the HuggingFace transformers library on a single NVIDIA A100 SXM4 80 GB GPU, executing the training for a total of 38,137 steps, which corresponds to approximately one epoch. During training, we employed a Linear Warmup strategy for the learning rate. The model was configured to handle a maximum token sequence length of 128 for both input and output, with a batch size set to 32. The fine tuning took 27 hours 8 minutes.

Fig. 2. Model training performance metrics over iterations. The three plots represent the changes in evaluation loss, training loss, and learning rate against the number of steps taken during the training phase of the model.

6 Results and discussion

This section presents the evaluation of our OCR correction model. We assess the model's performance using several metrics: Word Error Rate (WER), ROUGE-L score, and the identification of perfect matches in OCR erroneous sentences. Additionally, we analyze the model's capability to differentiate between erroneous and non-erroneous sentences. The evaluation was carried out by comparing the errors identified in the original text with the errors identified by the model in relation to the "target" (error-free) sentences. The test database contains a wide range of texts, from academic works to literature and newspaper articles.

6.1 Metric definitions

Before delving into the results, we define the metrics used for evaluation:

- WER (Word Error Rate): Measures the proportion of incorrect words to the total words in the reference text, lower values indicate better performance.
- ROUGE-L: Reflects the overlap of n-grams between the system output and reference texts, with higher scores indicating better quality.
- OCR Erroneous Sentences: Sentences identified by the model as containing OCR errors.
- Perfect Matches: Instances where the corrected text exactly matches the reference text.

6.2 The SOTA

Laki et al.'s [3] mT5 scored 0.923515 ROUGE-L on the test set (the same test set we used for the new model). The overall WER after correction was 0.224. (from 0.2327 = 0.9% improvement) Out of the 4799 sentences with OCR errors in the test set, only 198 have a perfect match between the corrected and the target sentence (4.13%). Their model incorrectly identified 60 out of 1981 non-erroneous sentences as erroneous, resulting in a false-positive rate of 2.98%.

6.3 Performance improvement

Our model demonstrates significant improvements in text correction accuracy, as evidenced by the metrics:

- The overall WER improved from 0.2327 to 0.1814, marking a 5.1% enhancement in the OCR erroneous sentences.
- For the entire test data, the improvement in WER is 0.148, amounting to a 6.5% improvement.
- The mean ROUGE-L score increased from 0.90 to 0.94 for OCR erroneous sentences, relative to the reference sentences.
- Out of 4799 OCR erroneous sentence pairs, 1095 were perfect matches after correction, achieving a 22.82% success rate.

– The model incorrectly identified 59 out of 1981 non-erroneous sentences as erroneous, resulting in a false-positive rate of 2.97%.

The observed improvements in WER and ROUGE-L scores highlight the effectiveness of our model in correcting OCR-generated text errors. The significant percentage of perfect matches further demonstrates the model's accuracy in identifying and correcting errors. However, the false-positive rate indicates a need for refinement in distinguishing between erroneous and non-erroneous sentences, suggesting an area for future work.

7 Conclusion

In our research aimed at correcting OCR errors, we efficiently employed the mT5 model, leveraging its Text2Text machine translation capabilities. We explored both mT5-small and mT5-large variants during the model's fine-tuning process. The outcomes suggest that the mT5 model is notably efficient in rectifying texts with OCR errors. We anticipate that improvement in the training dataset and the use of larger model variants could further improve correction accuracy. Additionally, we generated synthetic data to emulate OCR errors, thereby enriching our training dataset. The experimental results affirm the mT5 model's effectiveness in OCR error correction, highlighting the potential for achieving superior performance with ongoing advancements. Our review of relevant literature and international studies suggests that integrating character-based and sequence-to-sequence correction techniques could yield higher accuracy and reduce the likelihood of erroneous corrections. Moreover, the strategic application of Large Language Models (LLMs) in the detection, correction, and verification phases presents a promising direction for future research. The insertion of our recently developed gold standard corpus into the training data could also improve our results.

Acknowledgments. The present research was conducted with the support of the Hungarian Academy of Sciences in the framework of the Natonal Program 'Science in support of the Hungarian Language'.

References

1. Amrhein, C.: Post-Correcting OCR Errors Using Neural Machine Translation. Ph.D. thesis, Universität Zürich (2017), https://api.semanticscholar.org/CorpusID:231696696
2. Gupta, H., Del Corro, L., Broscheit, S., Hoffart, J., Brenner, E.: Unsupervised multi-view post-OCR error correction with language models. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 8647–8652. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.680, https://aclanthology.org/2021.emnlp-main.680

3. Laki, L.J., Kőrös, Á., Ligeti-Nagy, N., , Nyéki, B., Vadász, N., Yang, Z.Gy., Váradi, T.: OCR hibák javítása neurális technológiák segítségével [Correction of OCR errors using neural technologies]. In: XVIII. Magyar Számítógépes Nyelvészeti Konferencia. pp. 417–430. Szegedi Tudományegyetem, Informatikai Intézet, Szeged, Magyarország (2022), original text in Hungarian.
4. Maheshwari, A., Singh, N., Krishna, A., Ramakrishnan, G.: A Benchmark and Dataset for Post-OCR Text Correction in Sanskrit. In: Findings of the Association for Computational Linguistics: EMNLP 2022. pp. 6287–6294. Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.findings-emnlp.527, https://aclanthology.org/2022.findings-emnlp.527
5. Piotrowski, M.: Post-correction of OCR results using pre-trained language model (2021), http://poleval.pl/files/2021/09.pdf, presentation slides
6. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)
7. Rigaud, C., Doucet, A., Coustaty, M., Moreux, J.P.: ICDAR 2019 Competition on Post-OCR Text Correction. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1588–1593 (2019). https://doi.org/10.1109/ICDAR.2019.00255
8. Schaefer, R., Neudecke, C.: A Two-Step Approach for Automatic OCR Post-Correction. In: Proceedings of the Workshop on Computational Humanities Research (LaTeCH-CLfL 2020). pp. 52–57. Association for Computational Linguistics (2020)