LLM-based Pipelines for the Restoration of Digitized Historical Printed Archives

Arnaldo Tomasino[*], Saverio Ieva[*†], Floriano Scioscia[*†], Michele Ruta[*†], Ivan Lamparelli[‡], Rocco Michele Lancellotti[§]
[*] Polytechnic University of Bari – Via E. Orabona 4, Bari (I-70125), Italy – {name.surname}@poliba.it
[†] donkeyPower S.r.l. – Via E. Orabona 4, Bari (I-70125), Italy – {name.surname}@donkeypower.it
[‡] RCS Innovation S.p.A. – Prolungamento Viale Europa, 1/1° Traversa, Bari (I-70124), Italy – ivan.lamparelli@rcs.it
[§] RCS MediaGroup S.p.A. – via Rizzoli 8, Milan, Italy – michele.lancellotti@rcs.it

Abstract—**The digitization of historical archives enables not only preserving evidence of past events and of the evolution of society, but also opening up new digital content fruition opportunities. Unfortunately, Optical Character Recognition (OCR) tools, widely used for this purpose, may often produce erroneous text with older printed sources. Recently, the application of Large Language Models (LLMs) has been attempted to address this issue, but a systematic exploration is needed, given the ever growing variety of models and of prompt engineering techniques. This paper proposes two modular and scalable pipelines for structured invocation and evaluation of LLMs for the post-OCR text correction task. The goal is to build a general-purpose and extensible framework to be used in comparing the behavior of LLMs across various prompting approaches and configurations. The input dataset used for this purpose collects standardized OCR transcriptions of *Corriere della Sera* newspaper from the 19th and 20th centuries.**

Index Terms— **Large Language Models, prompt engineering, Optical Character Recognition, text correction, evaluation framework**

I. Introduction

The digitization of historical document collections, such as newspaper archives, has an important role in preserving cultural and social heritage and in building a complete and searchable corpus of historical information. For this purpose, Optical Character Recognition (OCR) tools are widely used to convert scanned images of historical newspapers into digital text. The resulting digital libraries are usually organized in XML files following well-established standards (*e.g.*, ALTO[1], METS[2], NIFT[3]). However, the results are heavily affected by factors such as degradation of original sources, low print quality, complex layouts, and different linguistic styles, resulting in noisy and error-filled text. The recent rise of Large Language Models (LLMs) has opened the opportunity to explore the application of these tools for the task of digitized historical text restoration, including text derived from newspapers [1] [2] [3] [4]. Post-OCR text correction can be considered as a *sequence-to-sequence* task [1] [2], a problem in which many LLMs have already been trained on. Several studies have analyzed the behavior of different LLMs on this task, with different combinations of prompts (*e.g.*, zero-shot, one-shot, few-shot approaches with and without context information), hyperparameters [5] and input data. OpenAI models have often been chosen due to their wide availability, while smaller models (*e.g.*, BART and Meta Llama variants) have been selected due to their training on denoising tasks [1]. Model performance is commonly evaluated using the Character Error Rate (CER), a metric based on the *Levenshtein Distance*. It considers the number of edits required to transform one string into another with substitutions, insertions and deletions. [1] [5] [6] The similar Word Error Rate (WER) metric, that operates at the word level, is also widely employed [7]. In [8], a modular framework for the evaluation of LLMs is proposed, with pipelines for dynamic ingestion of prompt templates, selection and configuration of models, input segmentation, post-processing and evaluation.

In this perspective, this work proposes two novel pipelines: (i) a LLM invocation pipeline, for automatic invocation of multiple models with various types of prompts; (ii) a LLM output evaluation pipeline to assess quantitatively the accuracy of the generated corrections. The structure of the paper is as follows: Section II describes the invocation pipeline and Section III the evaluation pipeline, before conclusion.

II. Invocation Pipeline

The proposed pipeline for LLM invocation supports different models and prompt types for OCR-generated texts by adopting a modular architecture. It can easily scale to handle large collections of OCR-generated texts, since it has been implemented without constraints on the size of the input dataset, as well as the number of models to be tested and the categories of prompts. The overall architecture is depicted in Figure 1.

The input dataset contains information extracted from XML files produced by OCR tools processing historical Italian newspapers. It includes the original XML filename, as a reference to the newspaper issue, text extracted through OCR scanning, manually corrected OCR text and the publication year of the source article. The dataset includes newspaper articles from the 19th and 20th century. The OCR-generated text in each entry of the dataset is dynamically embedded into the included prompt templates.

[1] https://www.loc.gov/standards/alto/
[2] https://www.loc.gov/standards/mets/
[3] https://iptc.org/standards/nitf/

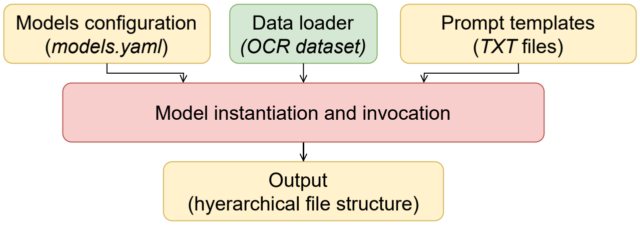

Fig. 1. Architecture of the proposed invocation pipeline

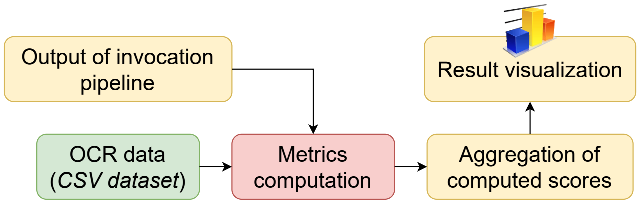

Fig. 2. Architecture of the proposed evaluation pipeline

External text files, stored in a dedicated input directory, contain different prompt templates. Here, additional textual instructions can be included to specify requirements that the generated output should meet (*e.g.*, in terms of formatting or preserving original characteristics of the source text). This structure allows easy addition of prompt template variations to be tested. Model configurations are stored in a separate YAML file. By modifying this file, new models can be added to the pipeline specifying their name and the platform on which they are available (*e.g.*, Hugging Face or OpenAI), without requiring modification of the source code. A dedicated function formats prompt templates with the appropriate OCR-generated text from the dataset and additional contextual information (*e.g.*, the year). Then, each model specified in the YAML file is invoked and the corresponding outputs are saved within a hierarchical directory structure organized by (i) the source file name (*i.e.*, the newspaper issue), (ii) the invoked model name and (iii) the prompt type.

III. Evaluation Pipeline

A. Evaluation Pipeline

Figure 2 illustrates the structure of the evaluation pipeline. It starts by reading the output files produced by the LLM invocation pipeline, keeping track of them in a structured representation that mirrors the previously generated directory hierarchy. For each entry there, evaluation metrics are computed by comparing the original OCR-generated text, the generated correction, and a reference correction. This reference correction can be either a manual correction already available in the dataset, or an automated correction generated by an additional LLM, acting as gold standard. Finally, the results are visually presented through grouped bar charts and plots, offering an intuitive summary of model performance across the different configurations.

In order to evaluate the accuracy of OCR-generated text corrections, the *CER reduction* metric [4] is employed:

$$CER\_reduction = \frac{CER(S,R) - CER(C,R)}{CER(S,R)} \qquad (1)$$

where S is the source, R is the reference and C is the LLM correction. Values closer to 1 indicates an improvement of the LLM-generated correction compared to the reference correction.

IV. Conclusion

This paper has introduced two modular pipelines based on LLMs for OCR-generated text correction. The invocation pipeline is extensible with additional LLMs and prompt approaches, systematically organizing the resulting output. The evaluation pipeline computes correction accuracy metrics (*e.g.*, CER) and enables visual comparison of model performance. Future work includes optimizing invocation pipeline execution and the adoption of input segmentation approaches to handle long texts with limited context windows. Subsequently, systematic experimental evaluations will be conducted using various LLMs across platforms and prompt template categories (including one-shot and few-shot approaches), clustering results w.r.t. these dimensions as well as other variables such as the historical period of the source material, in order to analyze the impact of individual features and select the best configurations.

Acknowledgments

This work has been supported by the *RCS Digital* grant (code EVY3OY8), co-funded by RCS Innovation S.r.l. and European Regional Development Fund for Apulia Region 2014/2020 Operating Program.

References

[1] A. Thomas, R. Gaizauskas, and H. Lu, “Leveraging LLMs for Post-OCR Correction of Historical Newspapers,” in *Third Workshop on Language Technologies for Historical and Ancient Languages*, 2024, pp. 116–121.

[2] E. Soper, S. Fujimoto, and Y.-Y. Yu, “BART for Post-Correction of OCR Newspaper Text,” in *Seventh Workshop on Noisy User-generated Text*. Online: Association for Computational Linguistics, 2021, pp. 284–290.

[3] L. Manrique-Gómez, T. Montes, A. Rodríguez-Herrera, and R. Manrique, “Historical Ink: 19th Century Latin American Spanish Newspaper Corpus with LLM OCR Correction,” 2024.

[4] V. Löfgren and D. Dannélls, “Post-OCR Correction of Digitized Swedish Newspapers with ByT5,” in *8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. Association for Computational Linguistics, 2024, pp. 237–242.

[5] J. Zhang, W. Haverals, M. Naydan, and B. W. Kernighan, “Post-OCR Correction with OpenAI’s GPT Models on Challenging English Prosody Texts,” in *Symposium on Document Engineering*. ACM, 2024, pp. 1–4.

[6] M. Veninga, “LLMs for OCR Post-Correction,” Master’s thesis, University of Twente, July 2024.

[7] S. de Araújo, B. Bezerra, A. Neto, and C. Zanchettin, “A proposal for post-OCR spelling correction using Language Models,” 2024.

[8] E. Boros, M. Ehrmann, M. Romanello, S. Najem-Meyer, and F. Kaplan, “Post-Correction of Historical Text Transcripts with Large Language Models: An Exploratory Study,” in *8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. Association for Computational Linguistics, 2024, pp. 133–159.