Communication Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 111–118
DOI: 10.15439/2025F4697
ISSN 2300-5963 ACSIS, Vol. 45

Opportunities and Challenges of LLMs as Post-OCR Correctors

Radoslav Koynov
0009-0003-8331-7475
Gesellschaft für wissenschaftliche Datenverarbeitung mbH
Burckhardtweg 4, 37077
Email: radoslav.koynov@gwdg.de

Triet Ho Anh Doan
0000-0002-7247-9108
Gesellschaft für wissenschaftliche Datenverarbeitung mbH
Burckhardtweg 4, 37077
Email: triet.doan@gwdg.de

***Abstract*—Large Language Models (LLMs) have demonstrated potential as zero-shot Post-OCR correctors for historical texts. However, previous research has typically focused on a single data set and only evaluated Character Error Rate (CER) or Word Error Rate (WER). This study investigates the potential of LLMs to enhance the accuracy of Optical Character Recognition (OCR) and the limitations of the models. To this end, an evaluation of the approach is conducted for a number of German and English historical datasets, with an in-depth analysis of the model corrections and deviation from the ground truth. We demonstrate that LLMs have the capacity to enhance the quality of OCR results as zero-shot correctors in some cases, and fine-tuning LLMs shows promise as part of an LLM-based Post-OCR correction system, if certain risks are mitigated carefully.**

I. Introduction

OPTICAL CHARACTER RECOGNITION (OCR) is the technology used to digitize printed and handwritten text, enabling large-scale text extraction from scanned documents. However, OCR systems are prone to errors, particularly when dealing with degraded documents, handwritten scripts, or complex layouts.

Recent advances in Large Language Models (LLMs) have opened new possibilities for automated post-OCR correction. LLMs, with their strong contextual understanding and ability to generate human-like text, offer a promising approach to refining OCR outputs by correcting errors and restoring missing characters. However, due to their nature as a generative model with a certain amount of creativity, they can also introduce new errors. These new errors may be qualitatively different from typical OCR errors, and potentially much harder to detect. Thus, the effectiveness and limitations of different LLMs, prompting techniques and fine-tuning strategies for post-OCR correction remain an open research area.

This paper presents a series of experiments that seek to enhance the accuracy of OCR texts through the utilization of LLMs. It is important to note that these models were not exposed to scanned images; rather, they were presented with OCR texts from various German and English datasets. The models were set up with a constant prompt and temperature during the course of the experiments. Furthermore, a fine-tuning process was implemented with the objective of enhancing the efficacy of the models.

In order to evaluate the results, a comparison was made between the Character Error Rate (CER) and Word Error Rate (WER) before and after the usage of LLMs. Additionally, we define the character change rate (CCR) and word change rate (WCR) analogously, but between the original OCR result as a reference, and the model-corrected version. A more in-depth examination is also conducted of the particular edit operations that are required in order to transform a piece of OCR text to its ground truth, and the edit operations that are implied by the LLMs.

II. Related Works

Prior research on the field of post-OCR correction has explored various models, datasets, and evaluation techniques to address errors in OCR-processed text.

Soper et al. [1] already show the capabilities of correcting noisy text outputs with pre-trained language models.

One of the more recent works on post-OCR correction using LLMs compares fine-tuned Llama2-7B, Llama2-13B, and BART on the BLN600 dataset, a collection of British newspapers from the 19^{th} centuries [2], [3]. They highlight the challenges posed by historical spelling conventions and employ a simple instruction prompt. This work impressively demonstrates the potential of fine-tuned LLMs for post-OCR correction, but does not sufficiently highlight certain risks or generalize to further datasets.

Earlier competitions, such as the ICDAR 2017 [4] and ICDAR 2019 challenges [5], provided foundational datasets for post-OCR correction. The ICDAR 2017 dataset includes 12 million aligned symbols extracted from newspapers and monographs in English and French, while the ICDAR 2019 competition expanded this effort to 22 million symbols across 10 European languages, focusing on multilingual post-OCR correction. Although these competitions predate modern LLMs, their datasets remain valuable for training and evaluation.

A notable study [6] in 2022 employed large ensembles of character sequence-to-sequence transformer models for post-OCR correction, achieving strong performance on the ICDAR 2019 dataset. This approach involved manually training a transformer model from scratch and segmenting documents into smaller pieces for processing. While effective, this method

requires extensive training and does not leverage the zero-shot or few-shot capabilities of modern LLMs.

Kanerva et al. use various LLMs for post-OCR correction on an English and a Finnish dataset and note that the results are much better for the English dataset [7]. Out of the models they employed, *GPT-4o* shows the most promise for both languages and achieves a reduction in the character error rate even for the Finnish texts. However, they conclude that this improvement for the Finnish texts is not considerable enough to practically attempt zero-shot post-OCR correction for Finnish at the present time.

A more recent study from 2024 evaluates OpenAI's *GPT-4*, *GPT-4 Turbo*, and *GPT-3.5 Turbo* models on post-OCR correction of challenging English prosody texts [8]. This study explores multiple prompts, metadata inclusion, and varying temperature settings. Using CER as the only metric, the study finds only marginal differences between models and prompt variations. While valuable, the dataset used in this study is not released to the community and highly specialized.

III. Datasets and Model Selection

A. Datasets

Three datasets were used in our experiment, as shown in Table I. BLN600 [3] contains English-language crime reports from newspapers from 1834 to 1894. This dataset has a relatively low initial error rate.

The next dataset was developed within the Optical Character Recognition Development (OCR-D) project, we refer to it as OCR-D-GT. Its content is based on transcription data stored in the German Text Archive [9]. The dataset is publicly accessible on GitHub [10], but contains only ground truth data. Therefore, for our experiments, an OCR workflow was executed on the text to produce OCRed texts. The workflow is straightforward and utilizes the `tesserocr-recognize` processor with the German Print [11] model.

Lastly, the ICDAR2019 dataset [5], introduced for the ICDAR 2019 Competition on Post-OCR Text Correction, comprises OCR outputs ground truth data for historical documents in multiple languages. We utilize the English and German subsets, which include digitized materials from sources such as the British Library and the German National Library. These texts contain a variety of printed materials, including newspapers and historical books. Specific publication years and genres are not detailed in the dataset's documentation.

All employed datasets contain ground truth data, i.e. documents already correctly digitalized by human experts which we use for evaluation of the results, as well as for preliminary fine-tuning experiments. They are structured in individual files. An OCR output file together with its ground truth file is referred to as a *page* or *document* throughout this work.

The average CER reported in Table I refers to what we measured for the full datasets.

TABLE I: Datasets for Post-OCR Correction

Dataset	Avg. CER	Pages	Language(s)	Years
BLN600	0.07248	600	English	1834-1894
OCR-D-GT	0.1486	217	German, others	1506-1897
ICDAR2019-EN	0.2018	150	English	
ICDAR2019-DE	0.2543	10,032	German	

B. Model Selection

In line with promising models from previous research, we select a model from the *Llama* family, and one from the *GPT* family.

GPT-4o mini [12] is optimized for efficiency while retaining strong multilingual reasoning capabilities, making it suitable for practical large-scale application as a post-OCR corrector.

Llama 3.3 70B [13] is a state-of-the-art instruction-tuned open-source LLM. It performs competitively on a range of benchmarks and is freely available for research and commercial use. It is also the successor of Llama 2, the model who showed promise for correcting errors in the BLN600 datasets.

IV. Metrics

We define several character-level metrics comparing the ground truth (GT), the original OCR output (OCR), and the model output (PostOCR) for a single document.

A. Character Error Rate

CER measures the minimum number of character-level edits (insertions, deletions, and substitutions) required to convert a hypothesis string into a reference string, normalized by the length of the reference:

$$\text{CER}(h, r) = \frac{S + D + I}{N} \tag{1}$$

where:

- h is the hypothesis string,
- r is the reference (ground truth),
- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- N is the number of characters in the reference.

Based on this, we define:

- $\text{CER}_{\text{old}} = \text{CER}(\text{OCR}, \text{GT})$: the error rate of the original OCR output against the ground truth.
- $\text{CER}_{\text{new}} = \text{CER}(\text{PostOCR}, \text{GT})$: the error rate of the model-corrected text against the ground truth.

B. Relative CER Reduction

To quantify the effectiveness of post-OCR correction, we define the relative improvement in CER as:

$$\textbf{CER Reduction} = \frac{\text{CER}_{\text{old}} - \text{CER}_{\text{new}}}{\text{CER}_{\text{old}}} \tag{2}$$

A value of 1 indicates perfect correction (i.e., all original errors were fixed), while a value of 0 indicates no improvement.

C. Character Change Rate

In addition to the traditional metrics comparing the OCR text with its ground truth, we introduce the**Character Change Rate (CCR)**. It quantifies the modification introduced by the post-OCR correction model, by using the original OCR output as the reference:

$$\text{CCR} = \text{CER}(\text{PostOCR}, \text{OCR}) \tag{3}$$

D. Change Ratio

From CCR, we derive a relative metric that quantifies the amount of change the model introduced with respect to the original CER.

$$\textbf{Change Ratio} = \frac{\text{CCR}}{\text{CER}_{\text{old}}} \tag{4}$$

A high Change Ratio together with a small CER Reduction indicates the model introduces many new errors.

E. Consecutive Edit Operations

While the CER and CCR capture the extent of changes necessary or introduced by a model, it does not reflect their distribution or locality. To address this, we define the *consecutive edit sequence* as a run of character-level edit operations that are adjacent in the edit space. We define adjacency according to the edit operations computed in the Levenshtein algorithm, whose output is an ordered list of tuples (op, i, j) where $\text{op} \in \{\text{I}, \text{S}, \text{D}\}$ at position i in the source string and j in the target. A sequence of operations is considered consecutive if the positions follow valid edit path transitions:

- `replace`: $(i+1, j+1)$
- `delete`: $(i+1, j)$
- `insert`: $(i, j+1)$

Given a threshold k, we compute the following metrics over these sequences:

- **Average Number of Consecutive Edit Sequences** $\geq k$ – The arithmetic mean of detected consecutive edit sequences $\geq k$ per document.
- **Average Length of Consecutive Edit Sequences** $\geq k$ – The average number of operations within each consecutive sequence meeting the threshold.

We also computed analogous metrics while restraining the types of consecutive operations, inspecting pure insertion and pure deletion sequences. This is useful to quantify missing information from OCR results as well as model hallucinations in post-OCR outputs.

F. Word-Level Metrics

All of these definitions are analogous at the word level, where insertions, deletions and substitutions are made at the level of words. For example, we use the term Word Change Rate (WCR) for the Word Change Rate, without defining it explicitly.

V. Experiment Setup

A. Data loading

We create a custom data loader for each of the datasets to homogenize the structure and prepare them to be passed to the LLMs for correction. The loader matches OCR texts with their ground truths to allow for automatic evaluation. The loaders use `dinglehopper` [14] to extract text from XML files (which are in PAGE [15] or ALTO [16] format), or plain-text files and apply minimal pre-processing.

B. LLM-based Post-OCR Correction

For each of the datasets, we run the LLM-based post-OCR correction pipelines, using the prompts shown in Figure 1. For the fine-tuned GPT models, a $75/25$ train-test split is used, and the evaluation results are reported on the documents of the test partition. Fine-tuning is done using OpenAI's fine-tuning API [17] with the default settings. Note that this does not include any holdout or cross-validation, but simply runs for a fixed number of epochs. We plan on implementing more sophisticated fine-tuning approaches with in future research.

Zero-Shot:

```
"You are a Post-OCR corrector. You
correct mistakes in historical
texts that are caused by errors in
the Optical Character Recognition.
You should NOT fix grammar or
spelling which deviate from Standard
{{language}}, because the texts are
historical. Please only include the
processed text in your response."
```

Fine-Tuned Models:

```
"You are a Post-OCR corrector. You
correct mistakes in historical
texts that are caused by errors in
the Optical Character Recognition.
Please ONLY include the corrected
text in your replies."
```

Common User Query:

```
"Please correct OCR-related mistakes
in the following historical text:
\n\n [OCR TEXT]"
```

Fig. 1: Prompt Templates Used for Post-OCR Correction

We use all models via a REST API and we use a temperature of 0.5 across all experiments for simplicity.

C. Automatic Evaluation

For each correction run, an automated evaluation script computes all metrics described in the previous section on a per-document basis and saves them as a dataframe. Additionally,

aggregations such as averages are computed and reported. The exact edit operations and a number of visualization plots are also saved automatically for each run.

VI. Evaluation and Analysis

In this section, we will first perform the standard evaluation based on CER and WER, before diving deeper to also investigate what changes the models applied and which errors it could (not) correct.

A. CER and WER

1) BLN600 - An English low-error dataset: For the *BLN600* dataset, the CER and WER reduction are displayed in Table II. Both the *GPT-4o mini* and the *Llama-3.3-70B* models achieved a significant reduction in the average CER. The GPT model and the open-source Llama model reduced the CER by almost 58% and 48% respectively. The zero-shot approach with GPT-4o-mini thus slightly outpeforms the fine-tuned Llama 2 model tested on this dataset [2], while the newer-generation Llama model almost achieves the performance.

On the word level, the improvements are even more considerable, with both models reducing the WER by over 75%. This is a clear indication that the models were particularly effective at correcting words with just one or few errors. Remaining errors might be in part due to sequences with accumulated errors, where it is increasingly hard or impossible to reconstruct missing information.

To get a better view of the distribution of error rates, it is useful to look at Figure 2, which shows the CER and WER of each document before and after correction for the more effective zero-shot model *GPT-4o mini*. The results demonstrate that both the CER and WER can be substantially reduced for numerous documents, particularly those exhibiting low initial CER values. Conversely, for documents with high initial CER, it becomes more challenging for the model to correct.

The fine-tuned model achieved even higher reduction in both CER and WER, reducing the character errors by almost 65% on average. This is a further indication that the fine-tuned models show promise of further improvements, when the zero-shot approach already yields good results. However, it should be noted that this is not necessarily statistically significant given the smaller test size for the fine-tuning approach.

2) OCR-D-GT - A tricky German dataset: For the German-language OCR-D-GT dataset, the aggregated results are summarized in Table III. Unfortunately, the models could not reach reduction in the character or word error rate. In fact, the CER actually increased by at least 30%.

The fine-tuned model performed much worse on average. It is extremely volatile and introduced many mistakes and even hallucinated entire paragraphs for some of the articles, leading to a large increase in CER when taking the arithmetic mean. This can partially be attributed to the relatively small and extremely heterogeneous dataset, covering several centuries with different genres and a spread of base CER from 0.23% up to 78.31%. Some of the fine-tuning examples of OCR results with up to 95% WER encourage the model to hallucinate corrections.

The fine-tuned model did however reduce the CER for a larger share of articles than the base model. This suggests that a more involved fine-tuning approach together with a larger and improved dataset can still be a promising approach.

3) ICDAR2019: For the ICDAR 2019 datasets, we use the German and English subsets, removing alignment data in the data loading step. In the case of the English dataset, both *Llama 3.3 70 B Instruct* and *GPT-4o mini* achieve a small reduction of the WER, but also a small increase in the CER. For the German texts, *GPT-4o mini* achieves a slight reduction in the CER and WER, while the Llama model yields very poor results, increasing the CER by 40%.

Although both models showed much promise as zero-shot correctors when employed for the BLN600 dataset, this is unfortunately not the case for the more complex datasets with higher initial error rates.

B. Comparing OCR and Model Output – CCR and WCR

There are various commonly found OCR errors, such as misinterpreted characters, disjointed characters and problems with hyphenation [18]. These might be recognizable to readers due to the visual similarity that lead to the error. Unfortunately, LLM correctors can introduce new types of errors that might be more problematic. For this reason, it is not enough to simply investigate CER and WER when comparing model performance. For example, in an OCR text with 10% CER, let a post-OCR correction model *A* reduce the CER to 5% by performing the edit operations needed to eliminate 5% of errors (CCR of 5%). Let model *B* also reduce the CER to 5%, but with a CCR of 7%. In this case, model *A* should be preferred since it did not introduce any new, potentially more problematic errors. In Figure 3, the CER before and after correction, as well as the CCR, are displayed for the BLN600 dataset.

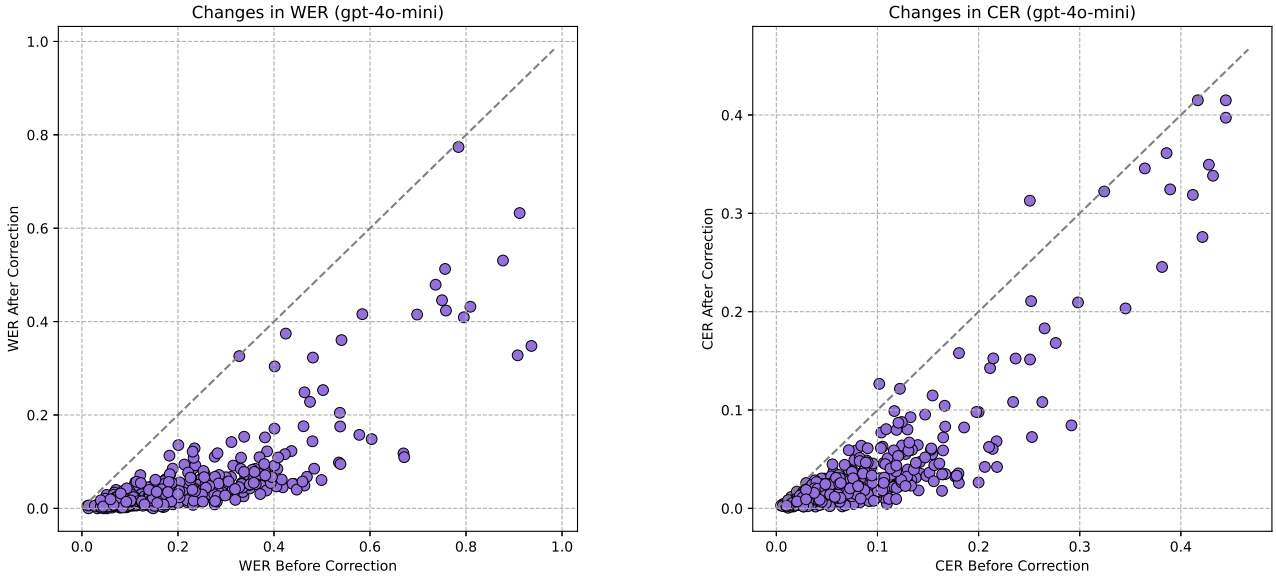
We can see that the *Llama 3.3 70B Instruct* model actually introduced more changes to the OCR text than *GPT-4o mini*, but unfortunately many of these changes did not reduce CER. On the other hand, it is a positive result that the fine-tuned model's higher CER reduction does not come with the price of an increased change rate.

As previously established, the results for the other datasets were not satisfactory. In the only other case, where a CER reduction was reached, the *GPT-4o-mini* model for the English-language subset of *ICDAR2019*, the model reduced the CER from 25.43% to 25.01% with a change rate of 3.92%. Although this is a relatively low CCR, given the amount of errors in the OCR result, it still means that the model introduced or changed existing errors amounting to more than 3% of the total characters, almost ten times more than it corrected.

As with the analysis of CER and WER, it is useful to gain a better view of the distribution of results on a document-by-document basis, instead of just considering averages. To visualize this, we add a color map to the scatter plot considered in the previous section. Since the CCR naturally correlates

TABLE II: Benchmarking LLMs for Post-OCR Correction on BLN600

Model	CER	WER	CER / WER Reduction
GPT-4o mini	0.07248 → 0.03065	0.18634 → 0.04404	**57.71%** / **76.37%**
Llama-3.3-70B	0.07248 → 0.03778	0.18634 → 0.04613	**47.89%** / **75.24%**
FT GPT-4o mini	0.06578 → 0.0231	0.16577 → 0.03216	**64.93%** / **80.6%**



(a) WER Scatter Plot

(b) CER Scatter Plot

Fig. 2: Per-Document Changes in WER and CER for BLN600 using GPT-4o mini.

TABLE III: Benchmarking LLMs for Post-OCR Correction on OCR-D-GT

Model	CER	WER	CER / WER Reduction
Llama-3.3-70B-Instruct	0.14855 → 0.17619	0.27290 → 0.36078	**-18.60%** / **-32.20%**
GPT-4o mini	0.14855 → 0.17735	0.27290 → 0.37705	**-19.39%** / **-38.16%**
FT GPT-4o mini	0.15716 → 0.37338	0.27118 → 0.53738	**-137.58%** / **-98.16%**

TABLE IV: Benchmarking LLMs for Post-OCR Correction on ICDAR-2019

Language Subset	Model	CER	WER	CER / WER Reduction
EN	Llama-3.3-70B-Instruct	0.20179 → 0.21304	0.31620 → 0.29992	**-5.57%** / **5.15%**
EN	GPT-4o mini	0.20179 → 0.20264	0.31620 → 0.31130	**-0.42%** / **1.55%**
DE	Llama-3.3-70B-Instruct	0.25430 → 0.35742	0.81175 → 0.83743	**-40.55%** / **-3.16%**
DE	GPT-4o mini	0.25430 → 0.25010	0.81175 → 0.77108	**1.65%** / **5.01%**

with both CER_{old} and CER_{new} it does not give a clear enough visual indication of the *relative* change. To account for this, we use the *Change Ratio* for the color axis, but clip the values at 2.0, which already indicates a very high change relative to the base CER, but keeps the scale readable at lower values.

In Figure 4, we can see that the Change Ratio for the BLN600 documents using *GPT-4o mini* is usually between 0.6 and 0.9, although it is lower for some documents that still exhibit a CER reduction. Concerning the documents with a high CER_{old} there are several documents with some CER reduction, but they generally have a significantly higher Change Ratio. There are also some documents with no improvements and barely any changes made by the model.

In Figure 5 the same plot is shown for the output of the *Llama 3.3 70B* model. We can see one outlier, where the model introduced a large amount of incorrect text for a single document. Apart from that, the plots look relatively similar, but due to the fixed color map with 0 on the low end and 2+ on the high end, we can also see that the Change Ratio is slightly higher for most documents, compared to the GPT model output.

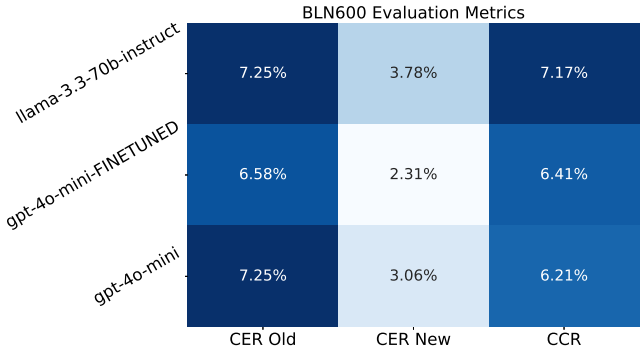


Fig. 3: Macro-averaged CER and CCR of correction models on BLN600

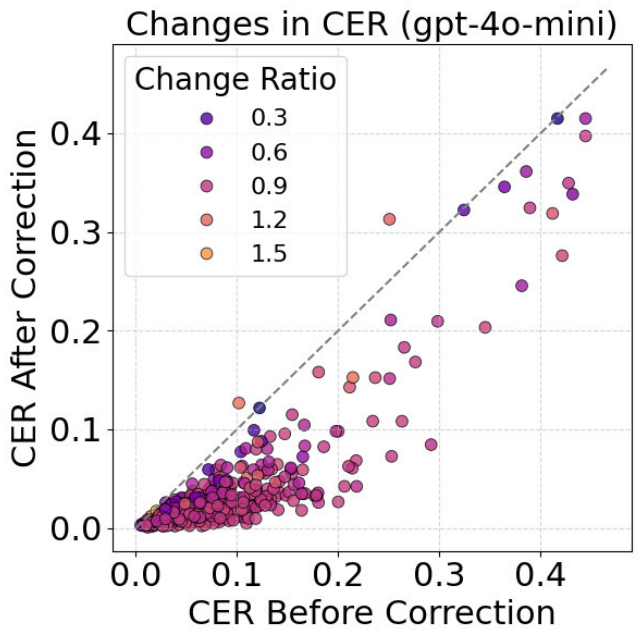


Fig. 4: Per-Document Changes in CER with Change Ratio as Color Axis for BLN600 using GPT-4o mini

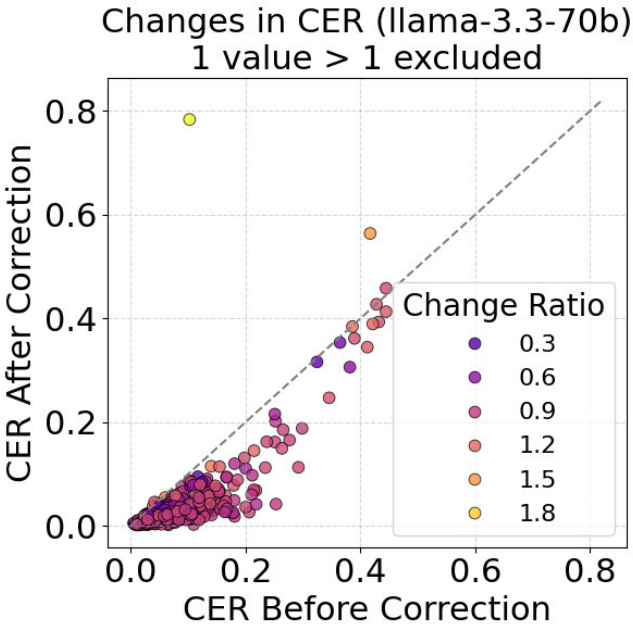


Fig. 5: Per-Document Changes in CER with Change Ratio as Color Axis for BLN600 using Llama 3.3 70B

C. Diving Deeper – Edit Operations and Consecutive Edits

When considering only the CER, the information about the types of edit operations in the shortest transformation sequence is lost. The average number of insertions (I), substitutions (S) and deletions (D) necessary to transform the OCR result to the Ground Truth (Expected) and to the Post-OCR document (Predicted) is given in Table V for all datasets. The position of the edit operations in the document can also be of interest. This is particularly the case when many errors occur consecutively in an OCR text, because this vastly increases the difficulty of the correction task. On the other hand, when a model prediction contains long consecutive sequences, especially of insertions, this is an indication of model hallucinations.

For BLN600, the dataset where the models achieved good results, we can see that the models predicted less insertions than were expected. *GPT-4o mini* predicted less insertions, but reached a higher CER reduction. This becomes clearer when we look at the consecutive edit sequences with minimum length $\boldsymbol{k = 6}$ operations. These are likely not retrievable from the OCR text. While *GPT-4o mini* applied 1.94 such sequences per document with an average length of 9.32, a large portion of these are pure "delete", on average namely 1.55 sequence per document with an average length of 9.89. This means that the model sometimes deleted sequences of characters that it deemed corrupted or illegible. On the other hand, it only applied 0.09 pure insertion sequences with an average length of 7.25.

For the same dataset, *Llama 3.3 70B* applied 2.87 consecutive edit sequences to such sequences with an average length of 10.49. While a considerable part of these were pure delete sequences as well, it also includes 0.26 pure insertion sequences of this length. This means that the Llama model's corrections contain some hallucinations, even for the dataset where we obtained a reduction in CER.

For the other datasets, the models do not achieve significant reduction of the error rate. Considering the expected edit operations and consecutive operations can give additional clues concerning the difficulty of the correction task for the various datasets. While the English-language ICDAR2019 and the OCR-D-GT data both have a higher number of expected consecutive operations, this is not the case for the German-language subset of ICDAR2019, which actually requires a large number of character substitutions, but few long consecutive transformation sequences. This means that missing information due to sequences of errors is not the sole reason

for unsuccessful LLM-based Post-OCR correction.

Unfortunately, standard fine-tuning with ground truth data encourages hallucinations instead of preventing them since ground truths contain coherent, legible text that can in some cases not be reconstructed from the OCR result alone. This holds true for both of the fine-tuned models we employed, as can be seen from the increased average number and length of consecutive edit sequences.

Taking a deeper look at pure insertions sequences, such "predicted" pure insertion sequences were usually of much greater length than arbitrary consecutive operations and especially prominent for the fine-tuned models, as well as the base *Llama* model for the German-language ICDAR2019 data. It predicted an average of 1.55 pure insertion sequences with an average length of 86.7 characters for this dataset.

While such insertion sequences indicate dangerous errors, they are easy to fix, once we are aware of them since they can easily be detected algorithmically without the need for a ground truth. Of course, picking a threshold and reverting insertion sequences above it, is a trade-off.

D. The Danger of LLMs as Post-OCR correctors – A concrete example

We have seen that models can introduce long sequences of characters to an OCR text. To show the effects of this, it is useful to consider an example. An excerpt from a BLN600 page with the ground truth, OCR result, and two model corrections are shown in Figure 6.

Both correctors fix the typical OCR error at the beginning of the excerpt, transforming "('harles" to "Charles". They both do not remove the hyphen for "Charles-street" seeing as the other names of streets are hyphenated in the text. They also both remove the duplicate "u" from "Trevor-squuare". Then, a passage with many errors starts. While the GPT model removes some of the characters, it manages to reconstruct some information and also keeps some illegible text. The Llama model on the other hand, is determined to create fluent legible text and hallucinates information for two full sentences, even introducing a new person, Mr. Miller, who is not mentioned anywhere in the OCR text. The model's training gives it a high incentive to create legible and grammatically correct text which outweighs the instructions to only correct OCR-induced error.

It should be mentioned that this excerpt is from one of the highest-CER documents of the BLN600 corpus. Although hallucinations of this scale are less likely in scenarios with lower base error rates, and some models are more prone to them than others, they can never be fully excluded.

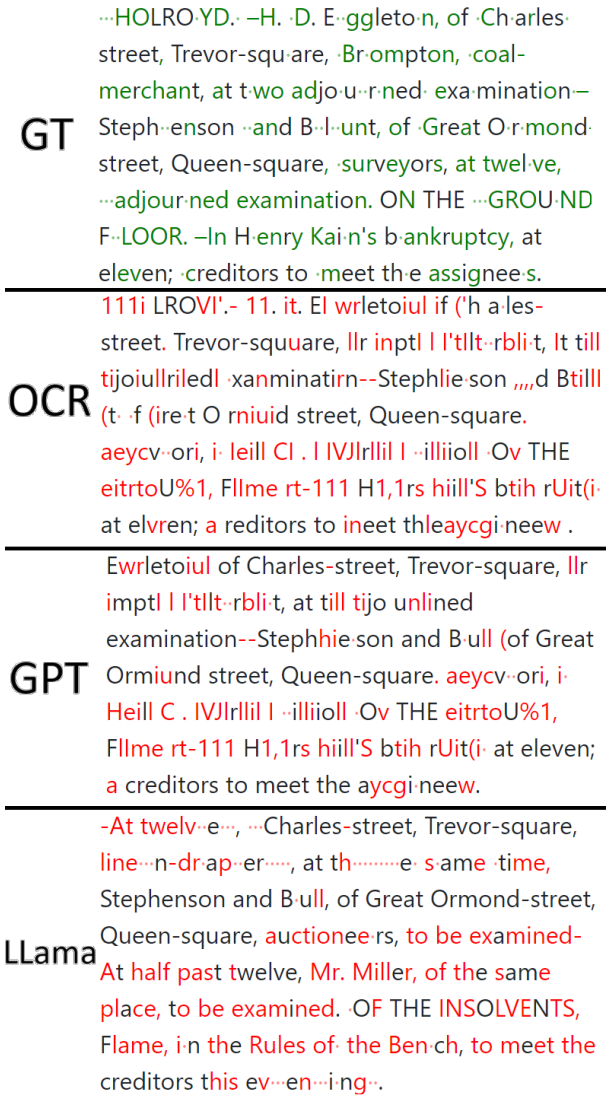


Fig. 6: Highly erroneous excerpt from BLN document – GT vs OCR vs GPT and Llama corrections

VII. Conclusion and Future Work

The paper presents the experiments in which LLMs were used in the post-correction step of an OCR workflow. It has been observed that when tasked with correcting errors in OCR texts, these models often introduce new and qualitatively different errors. However, the extent of these errors is relatively minor for certain datasets, particularly low-error English-language texts that don't deviate too from Standard English, such as BLN600. These errors can, however, be partially removed during post-processing. In addition, our preliminary findings indicate that fine-tuning significantly enhances model accuracy for the task of post-OCR correction, although it introduces additional risks such as overfitting and potentially increased hallucinations.

In subsequent studies, we intend to run LLMs locally to retain more control. Furthermore, different fine-tuning approaches will be tested, including utilizing synthetic datasets, which have demonstrated considerable potential in recent studies [19]. Particular focus will be placed on the development of a robust correction pipeline capable of consistently reducing OCR errors in historical texts, while simultaneously minimizing new model-induced errors.

TABLE V: Macro-averaged Expected and Predicted Edit Operations

Dataset	Model	I / S / D Operations	Avg. # Consec. Ops Diffs $\geq$ 6 chars	Avg. Consec. Ops $\geq$ 6 chars Length
BLN600	llama-3.3-70b-instruct	**Expected:** 39.06 / 79.76 / 76.58	2.24	11.89
		Predicted: 30.40 / 81.82 / 82.76	2.87	10.49
BLN600	gpt-4o-mini	**Expected:** 39.06 / 79.76 / 76.58	2.24	11.89
		Predicted: 19.97 / 68.16 / 82.15	1.94	9.32
BLN600	gpt-4o-mini (fine-tuned)	**Expected:** 48.12 / 73.80 / 64.33	2.32	13.11
		Predicted: 34.85 / 76.23 / 68.34	2.39	9.74
OCR-D-GT	llama-3.3-70b-instruct	**Expected:** 71.04 / 60.35 / 64.15	5.87	16.19
		Predicted: 18.28 / 43.80 / 33.30	1.16	14.37
OCR-D-GT	gpt-4o-mini	**Expected:** 71.04 / 60.35 / 64.15	5.87	16.19
		Predicted: 40.18 / 40.53 / 16.66	0.28	15.98
OCR-D-GT	gpt-4o-mini (fine-tuned)	**Expected:** 85.94 / 79.37 / 91.53	8.16	16.17
		Predicted: 559.94 / 164.76 / 53.82	22.67	26.80
ICDAR2019-DE	llama-3.3-70b-instruct	**Expected:** 41.83 / 255.78 / 85.49	1.06	11.16
		Predicted: 165.95 / 84.97 / 41.72	2.64	55.19
ICDAR2019-DE	gpt-4o-mini	**Expected:** 41.83 / 255.78 / 85.49	1.06	11.16
		Predicted: 13.10 / 31.61 / 18.16	0.29	17.83
ICDAR2019-EN	llama-3.3-70b-instruct	**Expected:** 112.50 / 124.32 / 107.65	9.17	19.60
		Predicted: 52.21 / 54.40 / 40.68	2.04	26.55
ICDAR2019-EN	gpt-4o-mini	**Expected:** 112.50 / 124.32 / 107.65	9.17	19.60
		Predicted: 19.44 / 39.07 / 42.53	1.23	18.92

REFERENCES

[1] E. Soper, S. Fujimoto, and Y.-Y. Yu, "BART for Post-Correction of OCR Newspaper Text," in *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, W. Xu, A. Ritter, T. Baldwin, and A. Rahimi, Eds. Online: Association for Computational Linguistics, Nov. 2021. doi: 10.18653/v1/2021.wnut-1.31 pp. 284–290. [Online]. Available: https://aclanthology.org/2021.wnut-1.31/

[2] A. Thomas, R. Gaizauskas, and H. Lu, "Leveraging LLMs for Post-OCR Correction of Historical Newspapers," in *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) @ LREC-COLING-2024*, R. Sprugnoli and M. Passarotti, Eds. Torino, Italia: ELRA and ICCL, may 2024, pp. 116–121. [Online]. Available: https://aclanthology.org/2024.lt4hala-1.14

[3] C. W. Booth, A. Thomas, and R. Gaizauskas, "BLN600: A Parallel Corpus of Machine/Human Transcribed Nineteenth Century Newspaper Texts," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, Eds. Torino, Italia: ELRA and ICCL, May 2024, pp. 2440–2446. [Online]. Available: https://aclanthology.org/2024.lrec-main.219/

[4] G. Chiron, A. Doucet, M. Coustaty, and J.-P. Moreux, "ICDAR2017 Competition on Post-OCR Text Correction," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017. doi: 10.1109/ICDAR.2017.232 pp. 1423–1428.

[5] Rigaud, Christophe and Doucet, Antoine and Coustaty, Mickaël and Moreux, Jean-Philippe, "ICDAR 2019 Competition on Post-OCR Text Correction," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019. doi: 10.1109/ICDAR.2019.00255 pp. 1588–1593.

[6] J. Ramirez-Orta, E. Xamena, A. Maguitman, E. Milios, and A. J. Soto, "Post-OCR Document Correction with large Ensembles of Character Sequence-to-Sequence Models," 2022. [Online]. Available: https://arxiv.org/abs/2109.06264

[7] J. Kanerva, C. Ledins, S. Käpyaho, and F. Ginter, "OCR Error Post-Correction with LLMs in Historical Documents: No Free Lunches," 2025. [Online]. Available: https://arxiv.org/abs/2502.01205

[8] J. Zhang, W. Haverals, M. Naydan, and B. W. Kernighan, "Post-OCR Correction with OpenAI's GPT Models on Challenging English Prosody Texts," in *Proceedings of the ACM Symposium on Document Engineering 2024*, ser. DocEng '24. New York, NY, USA: Association for Computing Machinery, 2024. doi: 10.1145/3685650.3685669. ISBN 9798400711695. [Online]. Available: https://doi.org/10.1145/3685650.3685669

[9] "Deutsches Textarchiv," https://www.deutschestextarchiv.de/, accessed: 2025-05-22.

[10] "gt_structure_text," https://github.com/OCR-D/gt_structure_text, Mar 2025, accessed: 2025-05-22.

[11] S. Weil, "Training German Print," https://github.com/UB-Mannheim/kraken/wiki/Training-German-Print, Jan 2024, accessed: 2025-05-22.

[12] OpenAI, "GPT-4o Mini: Advancing Cost-Efficient Intelligence," https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, 2024, accessed: 2025-05-16.

[13] M. AI, "LLaMA 3.3 Model Cards and Prompt Formats," https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/, 2024, accessed: 2025-05-16.

[14] M. Gerber and T. Q. S. Team, "Dinglehopper: An OCR Evaluation Tool," https://github.com/qurator-spk/dinglehopper, 2025, accessed: 2025-05-14.

[15] S. Pletschacher and A. Antonacopoulos, "The PAGE (Page Analysis and Ground-Truth Elements) Format Framework," in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 257–260.

[16] "ALTO Technical Metadata for Layout and Text Objects," https://www.loc.gov/standards/alto/, Jun 2022, accessed: 2025-05-23.

[17] OpenAI, "OpenAI Fine-Tuning API," https://platform.openai.com/docs/guides/fine-tuning, 2024, accessed: 2025-05-14.

[18] M. Lamba and M. Madhusudhan, "Exploring OCR Errors in Full-Text Large Documents: A Study of LIS Theses and Dissertations," *Library Philosophy and Practice (e-journal)*, no. 7824, 2023. [Online]. Available: https://digitalcommons.unl.edu/libphilprac/7824/

[19] J. Bourne, "Scrambled Text: Training Language Models to correct OCR Errors using Synthetic Data," 2024. [Online]. Available: https://arxiv.org/abs/2409.19735