MNLP 2025 - Homework 2

Thomas Chen and Thomas Garnier

1 Introduction

This project focuses on analyzing the cleaning of OCR-derived texts which generally contain errors due to input ambiguities. We used the OCRed version of The Vampyre by John William Polidori and processed them using 3 small LLMs in order to clean the text. The cleaning performance of each model was evaluated both by an LLM-as-a-judge and through manual annotations, enabling a comparison between machine and human assessments. We aim to (1) evaluate the reliability of LLMs-as-a-judge, and (2) investigate the benefits of combining OCR correction models to improve performance.

2 Methodology

Three different LLMs were used to correct the OCR texts : T5 tiny, BART and machine translation. Each LLM receives as input the noisy text and generates the corrected text as output. These corrections are then evaluated by an LLM-as-a-judge, with a prompt that specifies the grading scale defined by a strict grid (see in A.1). This LLM-as-a-judge provides a score from 1 to 5 for every corrected text. At the same time, we manually annotated the corrected texts, trying to follow the same notation grid. Then the distributions of human and LLM-assigned scores are compared in order to assess the relevance of the LLM-as-a-judge approach. Besides, we used the ROUGE-(1,2,L) metrics to evaluate the corrections and then we examined their consistency with LLM-based evaluation. Finally, we try to analyze the correlations across all these evaluation metrics.

To address the second question, whether combining LLMs specialized in OCR correction can improve performance.The OCR text is first processed by an initial LLM-based corrector, which generates intermediate corrected texts. These outputs are then passed as input to a second LLM-based corrector. The same evaluation procedure, as previously described, is applied to these final outputs.

3 Experiments

The first issue we encountered with all LLMs was their inability to generate outputs of unlimited sequence length. As a result, we had to split the text and apply the model at the sentence level rather than on the entire text. For sentence segmentation, we used the spaCy library with the `en_core_web` model.

3.1 T5

We selected T5 model because it's a text-to-text model pretrained on the huge C4 corpus. Therefore we assumed that T5 benefits from strong knowledge of grammar and syntax, enabling effective correction of OCR errors. However first results were very unsatisfying. Despite different prompt attempts, we saw that from the very first output sentences, results were pretty messy and not usable.

Due to the limited efficiency of T5-base, we switched to a grammar-specific variant: T5-base-grammar-correction. This model is a fine-tuned version of T5-base, trained specifically for grammatical error correction tasks. It showed noticeable improvements in text quality when applied to the first few sentences. However, it was too computationally heavy to apply to the entire corpus (3 min for 3000 characters), so we opted for a lighter and more efficient model even if it could reduce the performances: /t5-efficient-tiny-grammar-correction.

One last issue we encountered was that, since we prompted the model sentence by sentence, each output began with the word “corrected". Therefore we removed this perturbing word to clean the final text.

3.2 Back translation

This approach consists in using translation model that first translates into French and translates back

to English. This process may indirectly corrects OCR mistakes as the model focuses on fluency and adequacy. The limit of this method is that backtranslation could bring small changes, thus the output would not be completely aligned with the original sequence even if the meaning is preserved.

We tried two different models : MarianMT-Model and facebook/nllb-200-distilled-600M. We decided to choose MarianMTModel as the other one was significantly slower to run.

3.3 BART

BART was included in our experiments because it is specifically designed to denoise corrupted text sequences, making it highly relevant for our task. Comparing BART with T5 is particularly interesting, as both models share an encoder-decoder architecture. However, T5 is a large, general-purpose model while BART is more specialized in handling noisy text. This comparison allows us to evaluate whether such specialization offers a real advantage in this context.

3.4 Back translation + T5

Backtranslation alone did not fully remove OCR errors and sometimes altered the meaning of the text. To improve results, we combined back-translation (MarianMT) with T5 denoising. The idea is that the backtranslation corrects many errors through contextual translation, while T5 helps clean residual noisy words left untranslated. Although BART was more effective, T5 was chosen for its faster runtime. This method reduced errors, but sometimes at the cost of meaning preservation.

4 Results

Gemini was chosen for the LLM-as-a-judge because it is easy to integrate via its API and if freely available as a general-purpose model. As the number of tokens for each prompt was limited, we had to split the text into chunks. Each text was divided into chunks of lengths of 5000. The evaluation was performed on each chunk with the same prompt (see in A.2). As manual annotation is pretty time consuming, for each text, we annotated a fixed chunk (the 300 first characters) and two random picked chunks of size 300. After that, we averaged the score of the 3 chunks for each text with more weight on the fixed chunk (42%).

After computing the average scores for each text, we analyzed the results as follows: We applied ROUGE-1, ROUGE-2, and ROUGE-L metrics to compare each model's corrected text against the original clean version. The evaluation was performed on the first 20 sentences of each texts. Final scores for each model were obtained by averaging the ROUGE scores across all of the 48 texts. We can see the results on the table 1 bellow :

Method	ROUGE-1	ROUGE-2	ROUGE-L
Back-Translation	0.7214	0.4295	0.6409
BART	0.8547	0.7738	0.8430
T5	0.8296	0.7192	0.8185
OCR	0.8026	0.6577	0.8013
Back-Translation + T5	0.7226	0.4541	0.6388

Table 1: Scores ROUGE pour chaque méthode testée

We observe that the baseline ROUGE-1 score for the OCR method (around 0.80), which uses the raw OCR text without correction, is lower than the scores for the BART and T5 models, which reach above 0.82. This shows that these models improve the quality of the OCR output. However, the back-translation method performs worse than OCR, with a ROUGE-1 score near 0.72, indicating that it may introduce errors or produce different wording that reduces similarity to the reference. The combination of back-translation with T5 yields only a slight improvement over back-translation alone, suggesting that the combination method is not efficient to improve the quality of text.

We compare manual annotations with Gemini's judgments to assess if Gemini can reliably act as a judge. Section A.3 shows the score distributions for each model from both annotation sources. Figure 6 reveals that Gemini rates OCR texts higher (mean 3.55) than manual annotations (2.57), indicating an overestimation. Gemini's scores are also more concentrated, while manual annotations are more spread out. Gemini ranks models differently: back-translation lowest, BART highest, but oddly rates T5 below OCR. Manual annotations and ROUGE metrics, however, rank OCR worst, then back-translation, with BART clearly best. In summary, Gemini can identify the worst and best models but struggles with intermediate cases like T5. Additionally, Gemini's scores lack nuance, giving less extreme highs and lows compared to manual annotations (e.g., BART: 3.88 vs. 4.27; back-translation: 3.05 vs. 2.83).

A Appendix

A.1 Criteria

Score	Criteria
1	The text remains largely unreadable. Numerous errors persist (spelling, grammar, punctuation). The overall meaning is lost or extremely unclear. Little to no improvement made.
2	Slightly more readable than the original, but many errors remain. Several sentences are incorrect or ambiguous. Words are still distorted or missing.
3	Most obvious errors have been corrected. The text is generally understandable, but noticeable mistakes and awkward phrasing remain. The flow may be choppy.
4	The text is readable and coherent. Only minor errors remain. There's overall consistency, though a few syntax or word choice issues may still be present.
5	The text is fully corrected: no detectable mistakes, perfect grammar, punctuation, and syntax. The result is fluent, natural, and faithful to the original content.

Table 2: Evaluation criteria for corrected OCR outputs

A.2 Gemini Prompt

The prompt given to Gemini for the annotation of each chunk is : """You are a text quality evaluator. Evaluate the following corrected OCR text based on the detailed criteria below:

1 - Very Poor: The text remains largely unreadable. Numerous errors persist (spelling, grammar, punctuation). The overall meaning is lost or extremely unclear. Little to no improvement made. 2 - Poor: Slightly more readable than the original, but many errors remain. Several sentences are incorrect or ambiguous. Words are still distorted or missing. 3 - Fair: Most obvious errors have been corrected. The text is generally understandable, but noticeable mistakes and awkward phrasing remain. The flow may be choppy. 4 - Good: The text is readable and coherent. Only minor errors remain. There's overall consistency, though a few syntax or word choice issues may still be present. 5 - Excellent: The text is fully corrected: no detectable mistakes, perfect grammar, punctuation, and syntax. The result is fluent, natural, and faithful to the original content.

Give only the numeric score (1 to 5). No explanation.

Corrected Text: {chunk} """

A.3 Score Density

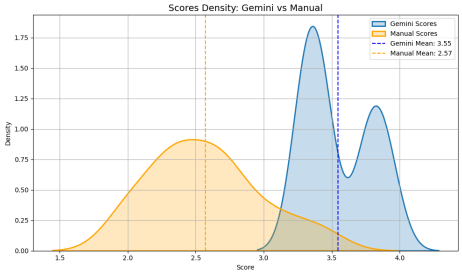

Figure 1: Score density on OCR

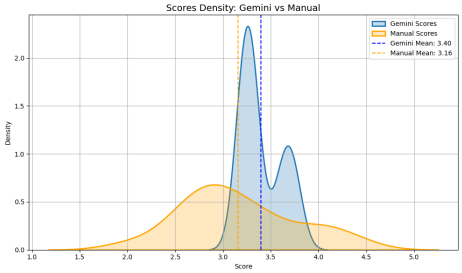

Figure 2: Score density on T5

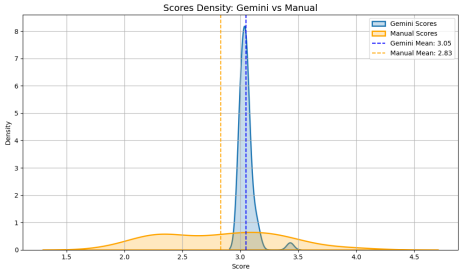

Figure 3: Score density on back translation

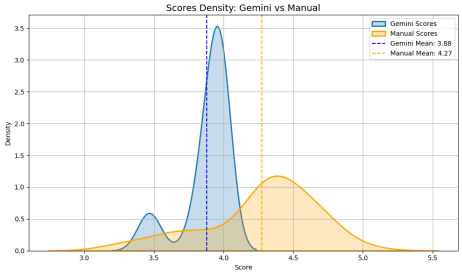

Figure 4: Score density on BART

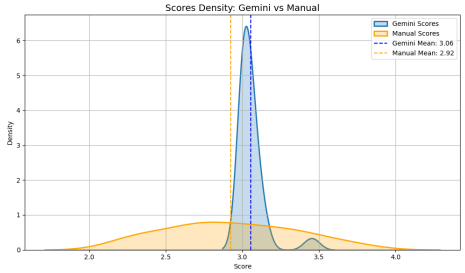

Figure 5: Score density on back translation + T5

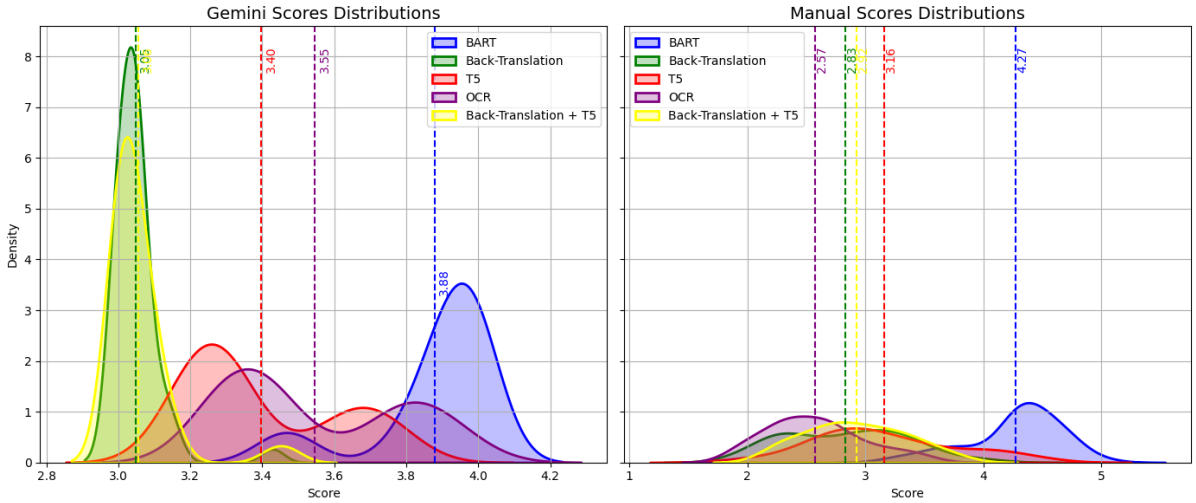

Figure 6: Resume : score density Gemini vs manual

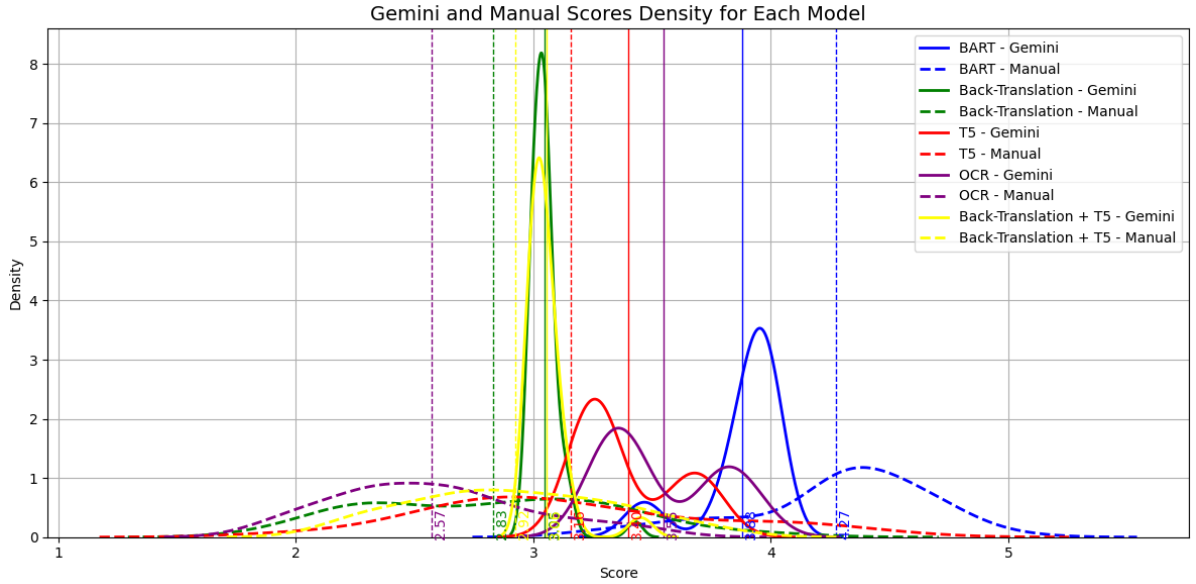

Figure 7: Resume : score density gemini and manual