Post-OCR Correction Using Large Language Models with Constrained Decoding

IGNACIO SASTRE, LORENA ETCHEVERRY, GUILLERMO REY, GUILLERMO MONCECCHI, and AIALA ROSÁ

Instituto de Computación, Facultad de Ingeniería, Universidad de la República (Udelar), Uruguay

This article addresses the problem of correcting noisy Optical Character Recognition (OCR) outputs from digitized historical documents, specifically those from the Berrutti Archive related to Uruguay's civic-military dictatorship. These documents—produced with typewriters, diverse layouts, and overlaid annotations—pose significant challenges for standard OCR tools, resulting in highly error-prone text. We present a novel post-OCR correction method that leverages fine-tuned open-source Large Language Models (LLMs) combined with a constrained decoding strategy. This strategy incorporates character-level similarity between the OCR input and the generated output at decoding time, steering the model toward corrections that closely preserve the original text structure. We evaluate our method on a gold-standard dataset of over 2000 annotated lines and show that it outperforms prompting and standard fine-tuning approaches, reducing both character error rate (CER) and word error rate (WER). The corrected outputs provide more accurate input for downstream tasks, such as named entity recognition, relation and event extraction, and knowledge graph construction, thereby supporting the broader goal of extracting knowledge from historically significant and sensitive archives.

Additional Key Words and Phrases: Natural Language Processing, Optical Character Recognition (OCR), Post-OCR Correction, LLMs with Constrained Decoding

1 Introduction

Between 1973 and 1985, Uruguay was governed by a civic-military dictatorship, preceded by a period of state terrorism from 1968 to 1973. This era was marked by systematic human rights violations, including torture, rape, kidnapping, and the forced disappearance of hundreds of individuals. Investigating these crimes has been historically hindered by legal impunity and the deliberate restriction of access to state-produced information.

In recent years, partial access to archival material has become possible. One notable example is the Berrutti Archive, which contains approximately three million pages of documents produced by security agencies during and after the dictatorship. The archive consists primarily of digital scans of microfilm reels, preserving heterogeneous and often degraded documents such as interrogation reports, press clippings, personal records, photographs, and lists of individuals and locations (Figure 1 shows some examples). These materials are of high historical and judicial value: they support ongoing investigations, contribute to the search for missing persons, and play a key role in fostering public understanding of this period.

Since 2018, we have been working on a multidisciplinary initiative [8] to analyze and systematize documents produced during Uruguay's authoritarian rule. A central challenge in this effort is extracting machine-readable text from scanned documents. Due to legal restrictions, the archival material cannot be processed using cloud-based services that require data sharing. As a result, we rely on on-premise OCR solutions such as Calamari OCR [33] and Tesseract[1]. To improve OCR performance, we built a proprietary gold-standard dataset via a crowdsourcing platform, where volunteers transcribe small document segments, which are later reviewed and curated by experts.

[1]Tesseract OCR https://github.com/tesseract-ocr/tesseract

Authors' Contact Information: Ignacio Sastre, isastre@fing.edu.uy; Lorena Etcheverry, lorenae@fing.edu.uy; Guillermo Rey, grey@fing.edu.uy; Guillermo Moncecchi, gmonce@fing.edu.uy; Aiala Rosá, aialar@fing.edu.uy, Instituto de Computación, Facultad de Ingeniería, Universidad de la República (Udelar), Uruguay.

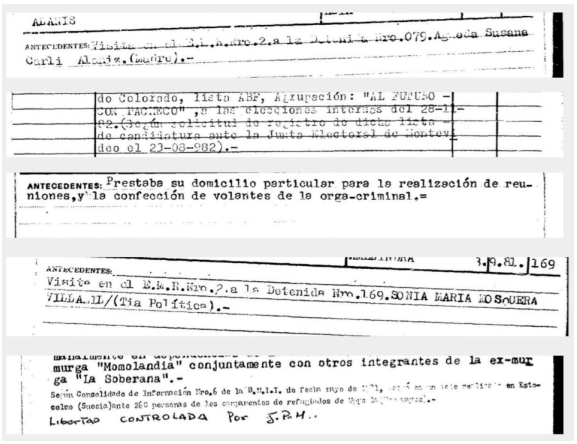

Fig. 1. Examples of document excerpts from the Berrutti Archive illustrating differences in the legibility and quality of the documents.

Despite these efforts, OCR outputs remain highly noisy due to the poor quality and variability of the source material. Errors at the character and word level significantly limit the usability of the extracted text, motivating the need for robust post-OCR correction methods.

In this paper, we address OCR post-correction as a sequence generation problem, where a noisy OCR output is transformed into a corrected version. Large Language Models (LLMs) are a natural candidate for this task, as they have demonstrated strong performance in generative language tasks and can leverage rich linguistic knowledge to resolve ambiguities [24, 27, 34]. However, standard LLM generation is not directly suitable in this setting. In particular, LLMs may introduce hallucinations or deviate from the source text—behaviors that are unacceptable in this domain, where preserving fidelity to the original document is essential for both historical accuracy and legal reliability.

We therefore frame OCR correction as a *controlled generation problem under fidelity constraints*. While constrained decoding has been successfully applied in tasks such as code generation and semantic parsing—where constraints enforce syntactic or structural validity—our setting presents a different challenge. Rather than enforcing a predefined output structure, we aim to ensure that the generated text remains closely aligned with a noisy input sequence, preserving its content while correcting errors. This requires controlling the generation process using criteria based on similarity to the input, rather than task-specific grammars or formal constraints.

To this end, we propose an approach that combines open-source LLMs fine-tuned on domain-specific examples with decoding strategies designed to guide generation toward outputs that remain close to the original OCR sequence. In particular, we explore (i) standard beam search to improve generation quality, and (ii) constrained beam search, which incorporates similarity-based constraints to enforce alignment with the input at the token level. Figure 2 illustrates this process, comparing the outputs produced by a fine-tuned LLM, a beam search variant, and a constrained beam search variant. The example highlights how different decoding strategies affect the trade-off between correcting OCR errors and preserving fidelity. The fine-tuned (FT) model produces outputs that are distant from the target text, whereas the two methods that leverage input distance yield significant improvements. Notably, although the FT model's output is erroneous, it includes a domain-specific entity —a real minister's name— demonstrating knowledge acquired through fine-tuning.

Contributions. In this paper, we:

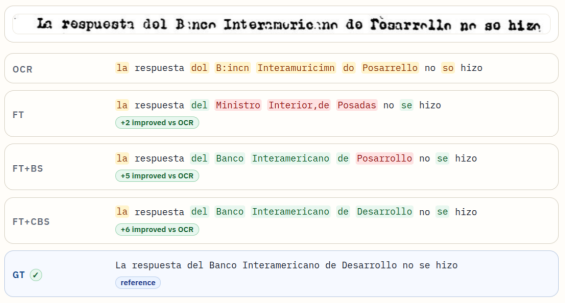

Fig. 2. Example of noisy OCR correction using an LLM. The OCR output introduces errors, which are corrected by the different versions of the LLM: the fine-tuned LLM (FT), the beam-search variant (FT+BS), and the beam-search variant with constrained decoding (FT+CBS). The different variants aim to maintain fidelity to the gold standard (GT). OCR errors are marked with yellow font, soft red text indicates modified, still wrongly translated text, and green text corresponds to Ground Truth values

- Propose an LLM-based approach to OCR post-correction that explicitly balances error correction with fidelity to the original text, a critical requirement in cultural heritage and human rights archives.
- Introduce similarity-based constrained decoding strategies that guide generation using alignment with the input sequence, at both sequence and token levels.
- Adapt controlled generation techniques to the challenging setting of noisy OCR text, demonstrating their effectiveness in reducing hallucinations.
- Evaluate our approach on a real-world archival dataset from a sensitive historical domain, highlighting practical constraints such as domain-specific language and on-premise processing requirements.

The paper is structured as follows: Section 2 presents a review of related work. Section 3 describes Fine-tuning experiments. Section 4 explains the beam search and similarity metrics at the sequence-level approach. Section 5 presents constrained beam search: a beam search approach combined with similarity metrics at the token level. Section 6 presents results on validation and test sets, together with some examples showing the impact of the proposed methods on OCR output correction. Section 7 identifies some limitations of the presented work. Finally, Section 8 contains the conclusions and outlines future work.

2 Related Work

Concerning post-OCR correction, two important references are the competitions carried out at the International Conference on Document Analysis and Recognition (ICDAR) in its 2017 [3] and 2019 [22] editions. For these competitions, datasets containing outputs generated by OCR tools, with errors, and the corresponding correct texts were made available. In 2017, the most effective systems integrated statistical machine translation and neural machine translation approaches. In 2019, a BERT-based method [7] achieved the best results. More recently, in 2021, a post-OCR processing survey [19] reviewed different datasets and methods, including some crowdsourcing approaches to create data for post-OCR processing. Some of them display entire documents that require manual correction by users [5, 13], while others show only single words [4, 31]. These approaches are similar to our project, where we created a crowd-sourced dataset using a web application. However, in our case, segments with few words are shown to users for correction, and we add an image with a fragment of the text to provide more context.

As with most language processing problems, current work on post-OCR correction prioritizes the use of transformer-based language models. In [28], experiments are described on a dataset of 19th-century English newspaper articles. They fine-tuned two transformer-based models: BART [16] and Llama 2 (7B and 13B) [29], the latter showing the best results. The authors mention that the most challenging problems faced were correcting named entities and LLama's hallucinations. In [1], the authors evaluate fourteen LLMs, with different prompts, against various post-correction benchmarks, including different domains and transcription qualities. They conclude that LLMs, used in this setting, are not particularly good for correcting transcriptions, but they also show that guiding their output format leads to better results.

In [17], the T5 model [21] was fine-tuned for OCR error correction specifically in Hungarian scientific texts. This fine-tuning led to improvements in the Word Error Rate (WER) and Rouge metrics. However, a key distinction between their work and ours is that they started with sentences that had significantly lower error rates compared to ours —0.90 in Rouge and 0.23 in WER, as opposed to our scores of 0.505 in Rouge and 0.633 in WER. In another study [30], a T5 model was employed for correcting OCR errors. The author demonstrates that while effective results can be achieved with modern texts, the same cannot be said for the ICDAR2019 dataset, which comprises ancient texts. Additionally, experiments conducted using the Llama model revealed issues related to model hallucinations, particularly the generation of text that is not present in the original sequence.

A different approach is presented in [12]. The authors start from the outputs of several OCRs and combine them using different language models: three GPT models (GPT, GPT2, GPT2XL) [20] and a 3-gram model trained on Wikipedia. This method aims to control the generative potential of generative models. Experiments are conducted using both the original pre-trained models and the fine-tuned models.

In a very recent work [2], an approach based on fine-tuning LLMs using synthetic examples is presented. A character-level Markov corruption process is applied to the examples to simulate the noise present in the texts.

Our proposal for post-OCR correction is also based on LLMs, but constrains the model's generation so that the generated corrections remain close to the original sequence. The main idea of this proposal is partly similar to what is mentioned in [32] as Controlled Generation: approaches based on the combination of the distribution generated by the model with some external criterion.

The concept of Constrained Decoding has been applied to various tasks, including the generation of code or SQL queries [10, 23], as well as the generation of linguistic analyses for texts, such as parsing, semantic role labeling, and semantic parsing [6, 25]. In these applications, constraints specific to the type of output being generated are checked at each step of the generation process. Different methods for modeling these constraints include finite-state automata [6] and grammars [9, 10]. Constrained decoding requires access to model logits during the inference process. This means that while it can be implemented with open models, it is not feasible with larger commercial models. To address this limitation, [9] proposes using a commercial model to generate an initial output and then employing a smaller open model with constrained decoding to enforce specific restrictions on the output.

3 Fine-tuning LLMs

In this section, we describe the overall post-OCR correction pipeline using fine-tuned LLMs. Given an OCR-generated sequence as input, the model is trained to produce a corrected version that closely matches a human-validated ground truth. This approach aims to teach the model the linguistic patterns of the domain as well as the typical distortions introduced by OCR tools in the Berrutti Archive.

We opted to fine-tune open-source LLMs for the OCR correction task after initial experiments with 2-shot prompting yielded suboptimal results (the prompt template used in these experiments is provided in Appendix A). The following subsections describe the training dataset, present the results of the initial fine-tuning experiments, and discuss the challenges encountered, particularly the tendency of LLMs to produce semantically plausible but incorrect substitutions, which motivates the need to constrain their generative behavior.

We used a dataset comprising 21,611 lines of text generated from a subset of documents using Calamari OCR, which was trained on data specific to our domain. For each of these lines, we created the ground truth (GT) using a web application, as previously described in Section 1. To fine-tune the LLMs, we transformed all pairs of ⟨ OCR output, GT transcription ⟩ into a string formatted as follows:

```
### ORIGINAL: {OCR output}
### CORRECTED: {GT transcription}
```

This format explicitly separates the OCR output (ORIGINAL) and the corrected transcription (CORRECTED), making it easier for the model to distinguish between the input and target text. We used those examples to fine-tune the `Llama 2` model from Meta [29], employing cross-entropy loss as the loss function (the same loss function used during model pretraining). We explored both the 7 and 13 billion parameter versions to evaluate trade-offs between performance and resource usage.

For these experiments, we utilized the ClusterUY infrastructure [18], which, during the experiments, consisted of two servers equipped with NVIDIA A100 GPUs and 28 servers with NVIDIA P100 GPUs. Given our resource constraints, we were unable to run experiments with models larger than those mentioned previously. Additionally, we employed the Low-Rank Adaptation (LoRA) technique [14], in which the model weights are frozen, and trainable rank decomposition matrices are injected into each layer of the model architecture. This method reduces the GPU memory requirements significantly because a much smaller number of weights are updated.

3.1 Evaluation and Problems

We evaluated the fine-tuned model on a test set of 2,163 ⟨OCR output, GT transcription⟩ pairs. We evaluated the transcriptions using three metrics:

(1) Ratio: This metric measures sequence similarity by computing 2M/T, where T is the total number of elements in both sequences and M is the number of matching elements. It is calculated using the SequenceMatcher.ratio function, provided by the `difflib` Python library[2]. The result is a value between 0 and 1, with higher values indicating greater similarity.
(2) Character Error Rate (CER): This metric computes the Levenshtein distance (the minimum number of single-character edits needed to transform one sequence into the other) between two sequences. The CER is calculated by dividing the Levenshtein distance by the length of the longer sequence, ensuring the result lies between 0 and 1. A lower CER indicates better similarity between the sequences.
(3) Word Error Rate (WER): Similar to CER, WER computes the Levenshtein distance but operates at the word level instead of the character level. The result is normalized by the total number of words in the reference sequence, yielding a value between 0 and 1. A lower WER indicates fewer word-level errors.

[2]SequenceMatcher documentation is available here: https://docs.python.org/3/library/difflib.html#difflib.SequenceMatcher.ratio

We used these metrics to compare the ground-truth transcriptions with the original OCR sequences (baseline), as well as with the corrected sequences produced by the adjusted models. Table 1 presents the average results for each metric under three conditions: (1) no corrections applied, (2) corrected using the prompting method with the `Llama 2 13B` model without fine-tuning, (3) corrected using our fine-tuned `Llama 2 13B` model.

Method	ratio (↑)	CER (↓)	WER (↓)
Original	**0.730**	**0,314**	**0,633**
Prompting	0.673	0.389	0.687
Fine-tuning	0.727	0.321	0.821

Table 1. Results for prompting and fine-tuning methods.

The results presented in Table 1 indicate that while fine-tuning improved character-level metrics compared to prompting (Figure 3), the overall performance still fell short of the OCR sequences without correction. Notably, the Word Error Rate (WER) increased significantly after fine-tuning, suggesting that the model occasionally introduced considerable errors during the correction process.

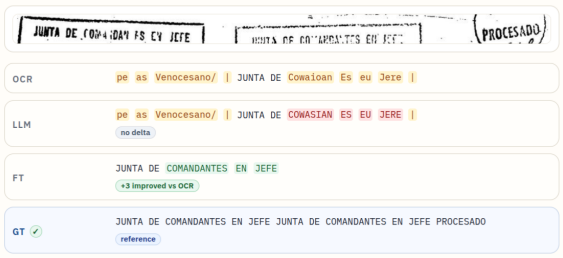

Fig. 3. Example of improvements by the fine-tuned model compared with the base model. While the prompting approach fails to correct the OCR output, the fine-tuned model successfully recognizes the entity "JUNTA DE COMANDANTES EN JEFE". This military terminology was clearly captured through domain-specific adjustment to our data. OCR errors are marked with yellow font, soft red text indicates modified, still wrongly translated text, and green text corresponds to Ground Truth values

We primarily observed that the model tended to replace original words with semantically plausible but incorrect substitutions. These included synonyms or similar terms that did not align with the ground truth. This behavior contributed to a higher WER due to mismatches at the word level.

Figure 4 illustrates this type of error, where the word *"sediciosa"* (seditious) is replaced with *"subversivas"* (subversive). Although these words are semantically similar, this type of solution is not suitable for the correction task, where the aim is to preserve the exact wording of the original text. Furthermore, the fine-tuned model exhibits hallucinations by generating text that is not related to the source input. Despite this, it generally incorporates domain-specific jargon, as seen in Figure 5, which features the expression "el anteriormente referido" (the aforementioned), which is a specific expression from our domain. This example clearly suggests that an approach that emphasizes fidelity to the original OCR output is necessary.

These problems highlight the need for methods that explicitly incorporate the original OCR sequence into the decoding process to guide the model during correction. In the following sections, we propose alternative decoding strategies that use character-level metrics to address this issue.

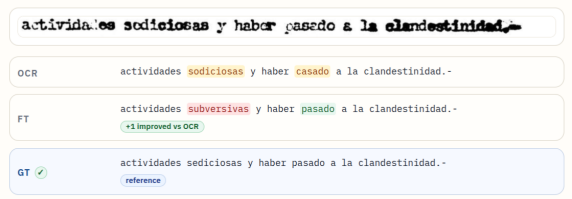

Fig. 4. Example of a semantically plausible but incorrect substitution (“subversivas” instead of “sediciosas”). OCR errors are marked with yellow font, soft red text indicates modified, still wrongly translated text, and green text corresponds to Ground Truth values

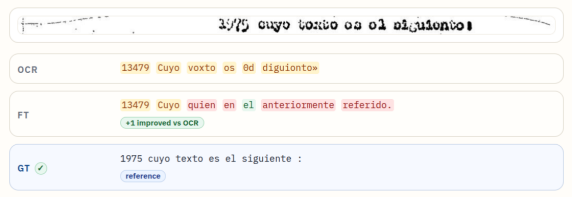

Fig. 5. Example of fine-tuned model hallucinations, where the invented text belongs to our domain (“el anteriormente referido”). OCR errors are marked with yellow font, soft red text indicates modified, still wrongly translated text, and green text corresponds to Ground Truth values

4 Beam Search with sequence-level similarity

Beam search [11, 26] is a decoding strategy used in probabilistic models, including LLMs, as an alternative to greedy decoding or sampling decoding. Greedy decoding often fails to produce optimal sequences because it selects the most probable token at each step without considering future choices, which can lead to missing better overall sequences.

In contrast, beam search explores multiple candidate sequences simultaneously by maintaining a “beam” of the top k most promising sequences at each step. During the decoding process, all possible next tokens from the vocabulary are considered for each of the k beams, resulting in $k \times V$ possible extended sequences, where V is the size of the vocabulary. Only the k most probable sequences are selected after each step to prevent exponential growth and continue expanding. At the end of this process, the sequence with the highest probability is chosen as the final output. This process generates a tree-like structure, as illustrated in Figure 6.

As mentioned in the previous section, the fine-tuned model encountered issues, notably the tendency to introduce semantically plausible but incorrect substitutions when employing a greedy decoding strategy. To address these problems, we explored the use of beam search with a subtle modification to the standard process.

Once the decoding steps are completed, we evaluate the k candidate sequences against the noisy OCR output by using the Character Error Rate (CER) instead of simply selecting the sequence with the highest probability. The sequence that has the smallest CER distance is then chosen as the final output.

This approach helps reduce the model’s tendency to replace original words with semantically similar but incorrect alternatives. By aligning more closely with the original sequence, this method significantly enhances performance, as demonstrated in Section 6.

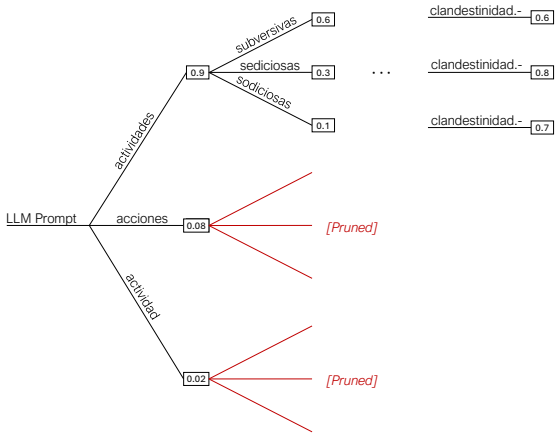

Fig. 6. Toy example illustrating beam search decoding with a beam width of three. Lower-probability branches (e.g., sequences derived from "acciones" and "actividad") are pruned to retain the three most promising candidate sequences at each step.

5 Constrained Beam Search with token-Level similarity

Although the previous method improved results across all metrics, the gains at the character level were modest compared to the OCR sequences. In contrast, the Word Error Rate (WER), which measures errors at the word level, remained worse than the OCR sequences on average (see Section 6 for details).

Since incorporating the CER distance at the end of Beam Search yielded noticeable improvements, we designed a Constrained Beam Search method to integrate character-level distance information throughout the decoding process of each step. Our approach incorporates character-level similarity not only at the end of decoding, but at each step of the sequence generation process.

The key idea is to guide the language model during decoding by combining its internal token probabilities with an external signal that measures how closely each partial sequence aligns with the original OCR input. By interpolating these two sources of information, the model is encouraged to make decisions that preserve fidelity to the input text.

Figure 7 illustrates this process: for each decoding step, the most probable tokens according to the language model are re-ranked using a similarity metric that compares the candidate partial output with the corresponding fragment of the OCR input. The result is a dynamic probability distribution that balances fluency and fidelity, effectively restraining the model's generative nature. The following paragraphs introduce the necessary notation to explain this method.

We will note the conditional probability of the next token w_i given the preceding sequence $w_1, w_2, ..., w_{i-1}$, as provided by the language model, as:

$$P_{LM}(w_i|w_1^{i-1})$$

The probability of a sequence $w_1, w_2, ..., w_n$ is then calculated as the product of the conditional probabilities of each token, given the preceding tokens:

$$P_{LM}(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} P_{LM}(w_i|w_1^{i-1})$$

For correcting the OCR output, we denote the noisy output of the OCR as s_1^n and the corrected sequence as w_1^m. In this context, we adapt the notation for the conditional probability of the next token w_i to explicitly include its

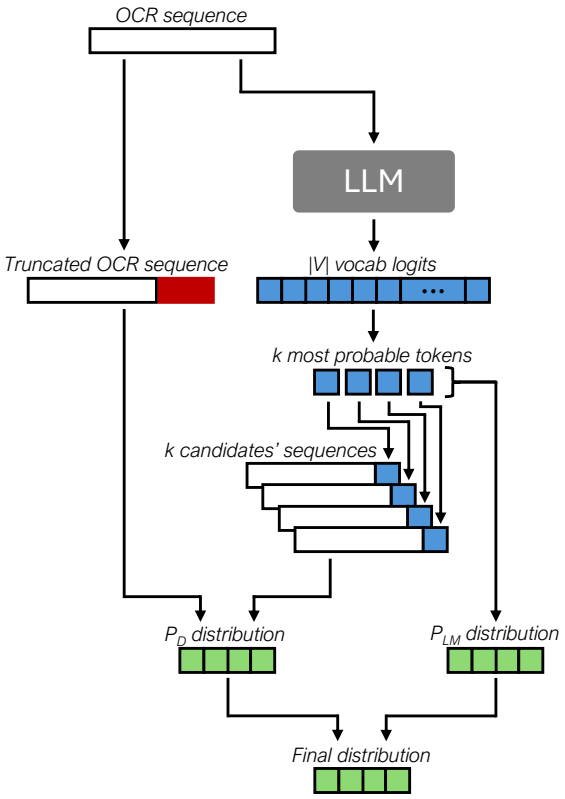

Fig. 7. Illustration of a step from the Constrained Beam Search (CBS) process. The k most probable tokens are selected and used for generating both P_D and P_{LM} distributions, which are then combined to obtain the final distribution.

dependence on s_1^n:

$$P_{LM}(w_i|s_1^n, w_1^{i-1})$$

We propose modifying the probabilities given by the language model at each step of the beam search process to account for the character-level distance between the generated sequence at that step and the noisy OCR output so far.

To achieve this, we define a probability distribution over the k most probable tokens at a given decoding step, denoted as P_D. This distribution aims to capture whether a token is likely to be correct by considering only the character-level distance between the generated sequence (including the candidate token) and the noisy OCR output. The distance is computed by comparing the generated sequence of i tokens, consisting of j characters, with the first j characters of the OCR output.

The probability distribution for the next token in the corrected sequence is obtained by interpolating between P_D and P_{LM}, where α is a hyperparameter to adjust:

$$\begin{aligned} P(w_i|s_1^n, w_1^{i-1}) \quad = \quad & \alpha P_D(w_i|s_1^i, w_1^{i-1}) + \\ & (1-\alpha) P_{LM}(w_i|s_1^n, w_1^{i-1}) \end{aligned}$$

This interpolated probability is used at each step of the beam search algorithm to adjust the sequence probabilities.

We apply the method described in the previous section to select the final sequence from the k candidate sequences generated at the end of the decoding process.

5.1 Calculating P_D

To calculate the P_D distribution, the Ratio metric is employed. As Section 3.1 explains, this metric returns the similarity between two sequences as a value between 0 and 1. To calculate $P_D(w_i|s_1^i, w_1^{i-1})$, the first step is to compute the Ratio between w_1^i and s_1^i, for each candidate token w_i. Assuming we have k candidate tokens $w_{i1}, w_{i2}, ..., w_{ik}$, we will compute k Ratios $r_1, r_2, ..., r_k$. After the Ratios are computed, the Softmax function is applied to obtain a probability distribution between all candidate tokens based on the previously calculated Ratios.

5.1.1 Dealing with close Ratios. Given that the candidate sequences being compared are generally very similar, the resulting Ratios are often close to each other and typically close to 1. For example, let's consider the following OCR output that needs correction (s) and two candidate corrected sequences, c_1 and c_2.

"actividades sodiciosas y haber casado a la clandestinidad.-" (s)

"actividades sediciosas y haber pasado a la clandestinidad.-" (c_1)

"actividades subversivas y haber pasado a la clandestinidad.-" (c_2)

The resulting Ratios are:

$$\text{SequenceMatcher}(c_1, s).\text{ratio}() = 0.966 \ (r_1)$$
$$\text{SequenceMatcher}(c_2, s).\text{ratio}() = 0.874 \ (r_2)$$

When applying the Softmax function, we get:

$$\text{Softmax}([0.966, 0.874]) \quad = \quad [0.5230, 0.4770]$$

The resulting probabilities are very close to each other, making the impact of P_D almost insignificant when the final interpolated distribution is calculated. However, c_1 and c_2 are rather different sequences, and this method's objective is to influence the closest sequence to the OCR output (in this case, c_1). To mitigate this problem, we explored using the temperature hyperparameter τ in the softmax function. To do so, we divide the Ratios by τ, where $\tau \in (0, 1]$ amplifies the probabilities of the closer candidates while diminishing those of the others [15]. When $\tau = 1$, the probabilities remain unchanged. As τ approaches 0, the distribution becomes more concentrated on the closest sequence, assigning a probability close to 1 to the closest sequence and near 0 to the others. This occurs because smaller values of τ amplify the differences between Ratios, and the softmax function tends to assign higher probabilities to higher values and lower probabilities to lower values.

Continuing with the previous example, applying a temperature of 0.1 results in probabilities $[0.715, 0.285]$. Reducing the temperature further, for instance to 0.05, makes the differences even more pronounced: $[0.863, 0.137]$.

5.1.2 Choosing a value for α. As we already explained, the final distribution is obtained by interpolating the distributions P_D and P_{LM}:

$$\begin{aligned} P(w_i|s_1^n, w_1^{i-1}) \quad &= \quad \alpha P_D(w_i|s_1^i, w_1^{i-1}) + \\ &\quad (1-\alpha) P_{LM}(w_i|s_1^n, w_1^{i-1}) \end{aligned}$$

We explored several methods for choosing the hyperparameter α. One straightforward approach is to choose a constant value for α. This value can be adjusted using a validation set. We identified a limitation of this static approach: we may not always want to assign P_D the same importance. More precisely, if the LLM is confident in one specific token over the others (i.e., the probability of a single token is much higher than the others), it would be convenient to prioritize the original model output even if the distance with the OCR output is high. On the other hand, if the LLM probabilities are not too conclusive (i.e., there are several tokens with non-negligible probabilities), then it would be desirable to prioritize the distance.

To better capture the level of uncertainty in P_{LM}, we explored using the entropy of the distribution to infer a value for α. In this approach, the value of α changes dynamically at each step of the beam search algorithm.

The entropy of the distribution is given by:

$$\begin{aligned} H(P_{LM}) \quad &= -\sum_{j=1}^{k} \quad P_{LM}(w_{ij}|s_1^n, w_1^{i-1}) \\ &\quad \log P_{LM}(w_{ij}|s_1^n, w_1^{i-1}) \end{aligned}$$

Since a value between 0 and 1 is needed, we normalize the entropy by dividing it by its maximum possible value, which is $\log k$. Therefore, α is defined as:

$$\alpha = \frac{H(P_{LM})}{\log k}$$

The intuition behind using entropy to define a dynamic value for α is based on using the LLM's confidence as a proxy for determining the relative importance of each distribution, since entropy measures the average level of uncertainty inherent in a probability distribution. If the entropy is high, α will be close to 1, and more importance will be given to the distance. If the entropy is low, α will be close to zero, and more importance will be given to the LLM probabilities.

In the following section, we present and compare the results obtained using both beam search with CER selection and the Constrained Beam Search method explained in this section.

6 Results

This section presents the results of all previously described experiments. Subsection 6.1 provides a detailed explanation of the results obtained on a validation set of 283 examples. It includes the outcomes for all experiments using different hyperparameter configurations, such as the temperature values and methods for calculating α, as discussed in the previous section. Subsection 6.2 reports the results on the test set described in Section 3.1, using only the best-performing configuration identified through the validation experiments. Finally, Subsection 6.3 shows some relevant examples of how the proposed methods improves the OCR output.

6.1 Results over the validation set

We conducted several experiments to select the best options among the alternatives discussed in Section 5. These experiments varied the transformations applied to the Ratio values (see Section 5.1.1) and the method for calculating the value of α (see Section 5.1.2).

In addition to using temperature, we explored two alternative transformations for the Ratio values: (1) Multiplying by 100, raising the result to a fixed exponent n, and normalizing by dividing by 100^n. (2) Assigning a probability of one to the highest Ratio while setting all others to zero. For the calculation of α, besides using entropy and a fixed value, we also experimented with a method using only the two largest probabilities, p_1 and p_2, defining α as $1 - (p_1 - p_2)$.

Table 2 presents the average results of the Ratio metric, including the counts of examples that improved, worsened, or remained equal compared to the original OCR sequences, and Table 3 reports the average results for word-level metrics, including WER, BLEU, and ROUGE. The last five rows in both tables show the results obtained by varying the temperature between 0.01 and 0.1. The best character-level performance on the validation set is achieved with a temperature of 0.05.

Based on the results from the validation set, we concluded that the best approach is to use temperature for transforming the ratios and a dynamic α calculated using entropy.

Ratio transform.	α	**Ratio (↑)**	**Better**	**Worse**	**Equal**
- (No correction)	*- (No correction)*	0.825	0	0	283
- (BS w/CER selection)	*- (BS w/CER selection)*	0.813	**127**	62	94
None	0.9	0.783	22	231	30
Raise to 10	0.5	0.826	86	110	87
None	$1 - (p_1 - p_2)$	0.833	123	43	117
Most similar	$1 - (p_1 - p_2)$	0.833	107	43	133
Raise to 2	$1 - (p_1 - p_2)$	0.834	117	42	124
Raise to 10	$1 - (p_1 - p_2)$	0.834	117	43	123
Raise to 2	Entropy	0.833	123	39	121
Raise to 10	Entropy	0.835	119	47	117
Temp = 0.01	Entropy	0.835	119	**31**	133
Temp = 0.03	Entropy	0.837	126	35	122
Temp = 0.05	Entropy	**0.838**	126	39	118
Temp = 0.07	Entropy	0.835	123	44	116
Temp = 0.1	Entropy	0.835	126	47	110

Table 2. Validation results using Ratio as metric.

6.2 Results over the test set

After selecting the best-performing method using the validation set, we evaluated it on the test set. Table 4 presents the average results of the Ratio metric, including the counts of examples that improved, worsened, or remained equal compared to the original OCR sequences. Table 5 provides the same analysis for the CER metric. Finally, Table 6 reports the average results for word-level metrics, including WER, BLEU, and ROUGE.

The first observation is that both methods improve performance across all metrics compared to the original OCR sequences without correction. The best results, on average, are obtained with the Constrained Beam Search (CBS) approach, with a temperature of 0.05, and the entropy-based dynamic method for selecting α. It outperforms other methods in both character-level metrics (Ratio and CER) and word-level metrics (WER and BLEU).

A closer analysis of the improved and worsened examples in the character-level metrics reveals that the Beam Search (BS) method with final CER selection produces more improved examples than the CBS method. However, CBS reduces the number of worsened examples and the number of unchanged examples. This observation suggests that the CBS

Ratio transform.	α	WER	BLEU	RougeL
- (No correction)	*- (No correction)*	0.486	0.412	0.642
- (BS w/CER selection)	*- (BS w/CER selection)*	0.582	0.482	0.721
None	0.9	0.436	0.284	0.523
Raise to 10	0.5	0.492	0.425	0.659
None	$1-(p_1-p_2)$	0.430	0.491	0.702
Most similar	$1-(p_1-p_2)$	0.448	0.477	0.692
Raise to 2	$1-(p_1-p_2)$	0.433	0.489	0.700
Raise to 10	$1-(p_1-p_2)$	0.433	0.489	0.700
Raise to 2	Entropy	0.429	0.494	0.707
Raise to 10	Entropy	0.431	0.490	0.703
Temp = 0.01	Entropy	0.423	0.501	0.714
Temp = 0.03	Entropy	**0.414**	0.514	0.724
Temp = 0.05	Entropy	**0.414**	0.515	0.725
Temp = 0.07	Entropy	0.413	0.517	0.730
Temp = 0.1	Entropy	**0.414**	**0.518**	**0.731**

Table 3. Validation results using word level metrics.

Method	Ratio (↑)	Better	Worse	Equal
Original	0.730	0	0	2163
Finetuning with Beam Search decoding and CER selection	0.735	**1339**	547	277
Finetuning with CBS decoding and and CER selection	**0.750**	1299	**462**	402

Table 4. Evaluation results using Ratio() as metric.

Method	CER (↓)	Better	Worse	Equal
Original	0.314	0	0	2163
Finetuning with Beam Search decoding and CER selection	0.307	**1278**	588	297
Finetuning with CBS decoding and and CER selection	**0.298**	1184	**533**	446

Table 5. Evaluation results using Character Error Rate as metric.

Method	WER (↓)	BLEU (↑)	RougeL (↑)
Original	0.633	0.26	0.505
Finetuning with Beam Search decoding and CER selection	0.776	0.356	**0.620**
Finetuning with CBS decoding and and CER selection	**0.508**	**0.421**	0.615

Table 6. Evaluation results using word level metrics.

method stays closer to the original OCR sequences, lowering recall (i.e., reducing the number of improved examples), but improving precision (i.e., reducing the number of worsened examples).

Figure 8 shows the distribution of examples across different Levenshtein distances (non-normalized CER), while Figure 9 presents a KDE plot of CER distances.

Both plots show that the number of perfect or nearly perfect sequences (i.e., low CER distance) doubles with both methods compared to the original OCR sequences. While BS with CER selection produces slightly more of these

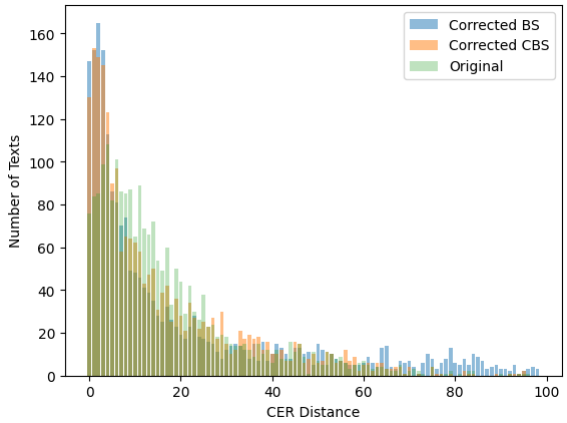

Fig. 8. Distribution of texts over different Levenshtein distances, using Beam Search and Constrained Beam Search decoding methods compared with the original noisy output.

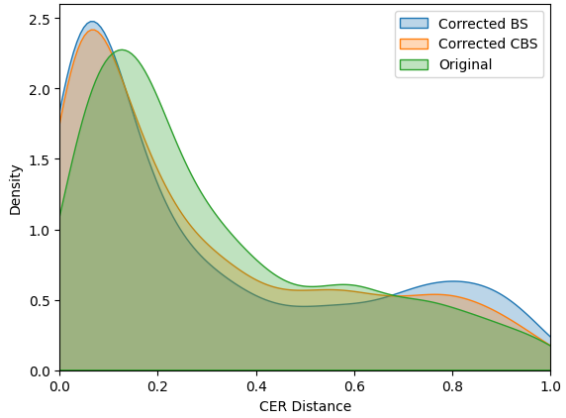

Fig. 9. Kernel Density Estimation (KDE) plot of CER distances, using Beam Search and Constrained Beam Search decoding methods compared with the original noisy output.

high-quality sequences, CBS significantly reduces the number of sequences with large distances, as observed in the graphs' tails.

It is important to note that there are many cases where the noise in the OCR sequence is so significant that reconstructing the ground truth sequence is impossible, even for a human. In such cases, the CBS method typically preserves the OCR sequence unchanged, whereas the BS with the CER selection method may produce entirely hallucinated outputs.

Finally, it is worth noting that the CBS method significantly reduces the WER distance. This observation confirms that CBS effectively mitigates the tendency to introduce semantically plausible but incorrect word substitutions.

6.3 Qualitative analysis

In this section, we present some additional examples of the impact of applying the methods proposed in this article on OCR results. Each figure shows, in order, the original image, the OCR output, the changes introduced by the fine-tuned model (FT), the impact of incorporating Beam Search (BS) and then Constrained Beam Search (CBS), as well as the human-annotated Ground Truth. We can observe that applying a domain fine-tuned LLM improves the OCR output, especially by incorporating domain elements (such as proper names or words related to the military jargon typical of these documents), and how the proposed adjustments to the generation process generally bring this output closer to the Ground Truth.

In Figure 10, the original scanned text is *"DOMICILIO: Pereyra Fuentes.- Fecha Afiliación: 20.1.69"*[3]. The original OCR has several problems, returning completely unreadable text. Using the fine-tuned LLM, the texts *"DOMICILIO:"* and "*Fecha de Afiliación:*" are correctly recognized. These expressions are highly recurrent in a specific type of document within our archive, namely cards with structured fields. Additionally, the date is adjusted to the format commonly used in the corpus. These changes are kept (with slight modifications) when incorporating the beam search procedures. We can also see that FT+CBS correctly translates the word *"Barrio:"* (a common continuation of *"DOMICILIO:"* in the corpus).

Fig. 10. Example of the LLM based methods improving the OCR output. OCR errors are marked with yellow font, soft red text indicates modified, still wrongly translated text, and green text corresponds to Ground Truth values

Figure 11 [4] shows how fine-tuned LLMs can incorporate domain elements into the transcription: in the early years of the dictatorship, the president of Uruguay was Juan María Bordaberry, and all the LLM-based models correctly adjust his surname, as well as several other words in the text. Interestingly, none of the models correct the word *"comitivo"* (which does not exist in Spanish), which, in this context, should be *"comitiva"*, demonstrating that there is still room for improvement.

In Figure 12 the original text is *"Ampliando lo informado en el Parte de Novedades de fecha de ayer, numeral 5"* [5] a phrase that likely appears more than once in the training corpus. Although the OCR output contains several errors, it can be seen that LLM post-processing, and especially the beam search methods, correct several of the words, almost completely identifying the text in the Ground Truth.

[3] *ADDRESS: Pereyra Fuentes.- Affiliation Date: 20.1.69"*
[4] *Mr. President of the Republic, Mr. Juan María Bordaberry and his entourage.*
[5] *Expanding on what was reported in yesterday's News Report, item 5,*

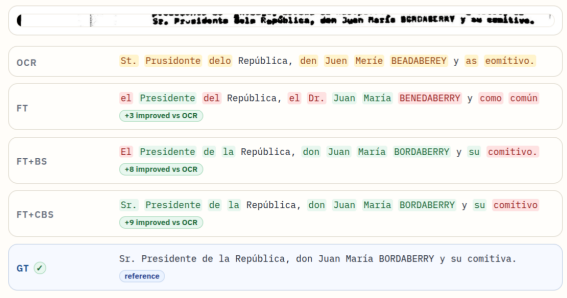

Fig. 11. Example of incorporating domain elements into transcripts. OCR errors are marked with yellow font, soft red text indicates modified, still wrongly translated text, and green text corresponds to Ground Truth values

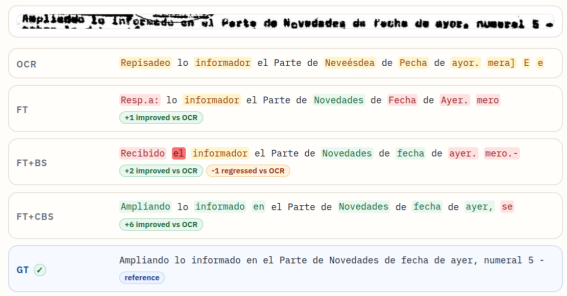

Fig. 12. Example of LLM post-processing adjusting several words in the OCR output. OCR errors are marked with yellow font, soft red text indicates modified, still wrongly translated text, and green text corresponds to Ground Truth values.

7 Conclusions and future work

In this work, we propose a method for correcting OCR outputs in the challenging domain of complex historical documents. Since documents contain sensitive information not publicly available, we are restricted from using cloud-hosted models, which limits our ability to compare results with leading closed-source LLMs.

Our approach combines fine-tuned large language models (LLMs) with a constrained decoding strategy. This strategy incorporates the character-level distance between the generated sequences and the original OCR sequences at each step of the decoding process, ensuring a closer alignment with the original text.

We observe improvements when comparing this method to traditional decoding techniques that rely on prompting and fine-tuning. The results indicate that our approach significantly reduces both the average character and word-level distances from the ground truth, greatly enhancing the accuracy of the OCR outputs.

In future work, we plan to extend this decoding strategy to multimodal models, such as `Llama 3.2`, which include vision capabilities. By integrating image information as additional context alongside the OCR output, we believe it will be possible to disambiguate sequences that are otherwise nearly impossible to correct.

We hope the findings in this work contribute to the major task of recognition and understanding of historical texts. The challenges posed by the Berrutti Archive are representative of those faced by many collections of documents of historical value. Our proposed solution offers a robust approach with potential for broad applicability in the field.

Acknowledgments

This work is part of the project “Inteligencia Artificial aplicada al procesamiento y la comprensión de archivos documentales”, founded by the Agencia Nacional de Investigación e Innovación (ANII) from Uruguay, and the International Development Research Centre (IDRC) from Canada. We would like to thank the members of the CRUZAR project[6] who contributed to the development of this work: Elina Gómez, Ignacio Ramírez, Fernando Carpani, Gregory Randall.

Data availability statements

Data supporting the findings of this study are openly available[7].

References

[1] Emanuela Boros, Maud Ehrmann, Matteo Romanello, Sven Najem-Meyer, and Frédéric Kaplan. 2024. Post-Correction of Historical Text Transcripts with Large Language Models: An Exploratory Study. In *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, Yuri Bizzoni, Stefania Degaetano-Ortlieb, Anna Kazantseva, and Stan Szpakowicz (Eds.). Association for Computational Linguistics, St. Julians, Malta, 133–159. https://aclanthology.org/2024.latechclfl-1.14/

[2] Jonathan Bourne. 2025. Scrambled text: Fine-tuning language models for OCR error correction using synthetic data - International Journal on Document Analysis And Recognition (IJDAR). *IJDAR* (2025). doi:10.1007/s10032-025-00522-0

[3] Guillaume Chiron, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2017. ICDAR2017 competition on post-OCR text correction. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 1. IEEE, 1423–1428.

[4] Otto Chrons and Sami Sundell. 2011. Digitalkoot: making old archives accessible using crowdsourcing. In *Proceedings of the 11th AAAI Conference on Human Computation (AAAIWS'11-11)*. AAAI Press, 20–25.

[5] Simon Clematide, Lenz Furrer, and Martin Volk. 2016. Crowdsourcing an OCR Gold Standard for a German and French Heritage Corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association (ELRA), Portorož, Slovenia, 975–982. https://aclanthology.org/L16-1155/

[6] Daniel Deutsch, Shyam Upadhyay, and Dan Roth. 2019. A General-Purpose Algorithm for Constrained Sequential Inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, Mohit Bansal and Aline Villavicencio (Eds.). Association for Computational Linguistics, Hong Kong, China, 482–492. doi:10.18653/v1/K19-1045

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. doi:10.18653/v1/N19-1423

[8] Lorena Etcheverry, Leopoldo Agorio, Virginia Bacigalupe, Sofía Barreiro, Elena Bing, Samuel Blixen, Daniel Calegari, Lautaro Cardozo, Fernando Carpani, Felipe Chavat, Diego Garat, Alvaro Gómez, Fabián Hernández, Victor Marabotto, Guillermo Moncecchi, Ignacio Ramírez, Aiala Rosá, Jorge Tiscornia, Dina Wonsever, Gregory Randall, Guillermo Zorron, Lia Rivero, Nilo Patiño, Javier Stabile, Ernesto Fernandez, Federico Fioritto, and Rodrigo Laguna. 2021. A computational framework for the analysis of the Uruguayan dictatorship archives. In *Proceedings of the Conference on Digital Curation Technologies (Qurator 2021), Berlin, Germany, February 8th - to - 12th, 2021 (CEUR Workshop Proceedings, Vol. 2836)*, Adrian Paschke, Georg Rehm, Jamal Al Qundus, Clemens Neudecker, and Lydia Pintscher (Eds.). CEUR-WS.org. https://ceur-ws.org/Vol-2836/qurator2021_paper_20.pdf

[9] Saibo Geng, Berkay Döner, Chris Wendler, Martin Josifoski, and Robert West. 2024. Sketch-Guided Constrained Decoding for Boosting Blackbox Large Language Models without Logit Access. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 234–245. doi:10.18653/v1/2024.acl-short.23

[10] Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 10932–10952. doi:10.18653/v1/2023.emnlp-main.674

[11] Alex Graves. 2012. Sequence Transduction with Recurrent Neural Networks. *CoRR* abs/1211.3711 (2012). arXiv:1211.3711 http://arxiv.org/abs/1211.3711

[12] Harsh Gupta, Luciano Del Corro, Samuel Broscheit, Johannes Hoffart, and Eliot Brenner. 2021. Unsupervised Multi-View Post-OCR Error Correction With Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing

[6] https://ceur-ws.org/Vol-2836/qurator2021_paper_20.pdf
[7] https://gitlab.fing.edu.uy/memoria/datasets-publicos/-/tree/main/luisa_tesseract_lines

Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 8647–8652. doi:10.18653/v1/2021.emnlp-main.680

[13] Rose Holley. 2009. *Many hands make light work: Public collaborative OCR text correction in Australian historic newspapers.* Technical Report. National Library of Australia. http://nla.gov.au/ndp/project_details/documents/ANDP_ManyHands.pdf

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models.. In *Tenth International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2106.09685

[15] Daniel Jurafsky and James H. Martin. 2025. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models* (3rd ed.). https://web.stanford.edu/~jurafsky/slp3/ Online manuscript released January 12, 2025.

[16] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 7871–7880. doi:10.18653/v1/2020.acl-main.703

[17] Gábor Madarász, Noémi Ligeti-Nagy, András Holl, and Tamás Váradi. 2024. OCR Cleaning of Scientific Texts with LLMs. In *Natural Scientific Language Processing and Research Knowledge Graphs*, Georg Rehm, Stefan Dietze, Sonja Schimmler, and Frank Krüger (Eds.). Springer Nature Switzerland, Cham, 49–58.

[18] Sergio Nesmachnow and Santiago Iturriaga. 2019. Cluster-UY: Collaborative Scientific High Performance Computing in Uruguay. In *Supercomputing*, Moisés Torres and Jaime Klapp (Eds.). Springer International Publishing, Cham, 188–202.

[19] Thi Tuyet Hai Nguyen, Adam Jatowt, Mickael Coustaty, and Antoine Doucet. 2021. Survey of Post-OCR Processing Approaches. *ACM Comput. Surv.* 54, 6, Article 124 (July 2021), 37 pages. doi:10.1145/3453476

[20] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. https://api.semanticscholar.org/CorpusID:160025533

[21] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2019), 140:1–140:67. https://api.semanticscholar.org/CorpusID:204838007

[22] Christophe Rigaud, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2019. ICDAR 2019 competition on post-OCR text correction. In *2019 international conference on document analysis and recognition (ICDAR)*. IEEE, 1588–1593.

[23] Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 9895–9901. doi:10.18653/v1/2021.emnlp-main.779

[24] Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024. The BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, Ekaterina Kochmar, Marie Bexte, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Anaïs Tack, Victoria Yaneva, and Zheng Yuan (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 571–589. https://aclanthology.org/2024.bea-1.51

[25] Richard Shin, Christopher Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained Language Models Yield Few-Shot Semantic Parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 7699–7715. doi:10.18653/v1/2021.emnlp-main.608

[26] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf

[27] Anaïs Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, Ekaterina Kochmar, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Nitin Madnani, Anaïs Tack, Victoria Yaneva, Zheng Yuan, and Torsten Zesch (Eds.). Association for Computational Linguistics, Toronto, Canada, 785–795. doi:10.18653/v1/2023.bea-1.64

[28] Alan Thomas, Robert Gaizauskas, and Haiping Lu. 2024. 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*, Rachele Sprugnoli and Marco Passarotti (Eds.). ELRA and ICCL, Torino, Italia, 116–121. https://aclanthology.org/2024.lt4hala-1.14

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi

Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL] https://arxiv.org/abs/2307.09288

[30] M.E.B. Veninga. 2024. LLMs for OCR Post-Correction. http://essay.utwente.nl/102117/ Master Thesis.

[31] Luis von Ahn, Benjamin Maurer, Colin McMillen, David J. Abraham, and Manuel Blum. 2008. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science* 321 (2008), 1465 – 1468. https://api.semanticscholar.org/CorpusID:18371056

[32] Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. 2024. From Decoding to Meta-Generation: Inference-time Algorithms for Large Language Models. *Transactions on Machine Learning Research* (2024). https://openreview.net/forum?id=eskQMcIbMS Survey Certification.

[33] Christoph Wick, Christian Reul, and Frank Puppe. 2020. Calamari- A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition. *DHQ: Digital Humanities Quarterly* 14, 2 (2020).

[34] Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual Machine Translation with Large Language Models: Empirical Results and Analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 2765–2781. doi:10.18653/v1/2024.findings-naacl.176

A Prompt used in the initial prompting experiments

Below we provide an English translation of the prompt used in our initial 2-shot prompting experiments. The original prompt was written in Spanish, since the OCR correction task was performed on Spanish text.

A.1 System prompt

```
You will receive a passage from a text produced by an OCR system and therefore containing noise.
Your task is to return the same text while correcting the noisy parts. Text that does not make sense
should be removed, and spelling mistakes should be corrected by approximating standard Spanish usage.
Process the full excerpt, from beginning to end, until the final closing quotation mark.
```

A.2 Few-shot format

The prompt included two user–assistant demonstration pairs. We show below the structure of the context as presented to the model.

First user message

```
– EXTRACT 1 –

### ORIGINAL:
"{OCR_input_example_1}"
```

First assistant message

```
### CORRECTED:
"{ground_truth_example_1}"
```

Second user message

```
— EXTRACT 2 —

### ORIGINAL:
"{OCR_input_example_2}"
```

Second assistant message

```
### CORRECTED:
"{ground_truth_example_2}"
```

A.3 Final input message

After the two demonstration pairs, the OCR output to be corrected is appended as a third user message, following the same format as the previous user messages.

Third user message

```
— EXTRACT 3 —

### ORIGINAL:
"{OCR_output}"
```