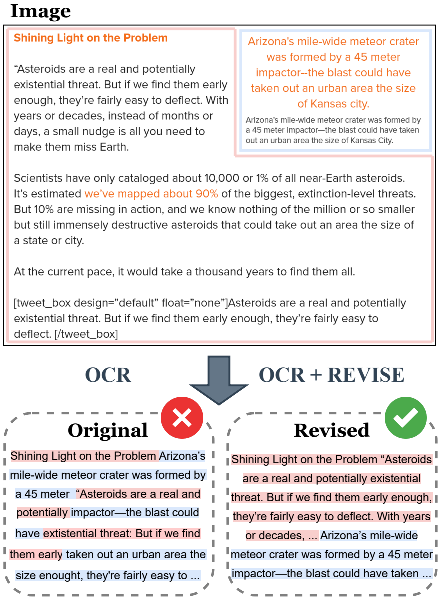
REVISE: A Framework for Revising OCRed text in Practical Information Systems with Data Contamination Strategy

Gyuho Shim[1]*, Seongtae Hong[1]*, Heuiseok Lim[1,2]†
[1]Department of Computer Science and Engineering, Korea University
[2]Human-inspired AI Research,
{gjshim, ghdchlwls123, limhseok}@korea.ac.kr

Abstract

Recent advances in Large Language Models (LLMs) have significantly improved the field of Document AI, demonstrating remarkable performance on document understanding tasks such as question answering. However, existing approaches primarily focus on solving specific tasks, lacking the capability to structurally organize and manage document information. To address this limitation, we propose REVISE, a framework that systematically corrects errors introduced by OCR at the character, word, and structural levels. Specifically, REVISE employs a comprehensive hierarchical taxonomy of common OCR errors and a synthetic data generation strategy that realistically simulates such errors to train an effective correction model. Experimental results demonstrate that REVISE effectively corrects OCR outputs, enabling more structured representation and systematic management of document contents. Consequently, our method significantly enhances downstream performance in document retrieval and question answering tasks, highlighting the potential to overcome the structural management limitations of existing Document AI frameworks.



Shining Light on the Problem

"Asteroids are a real and potentially existential threat. But if we find them early enough, they're fairly easy to deflect. With years or decades, instead of months or days, a small nudge is all you need to make them miss Earth.

Arizona's mile-wide meteor crater was formed by a 45 meter impactor--the blast could have taken out an urban area the size of Kansas city.

Arizona's mile-wide meteor crater was formed by a 45 meter impactor—the blast could have taken out an urban area the size of Kansas City.

Scientists have only cataloged about 10,000 or 1% of all near-Earth asteroids. It's estimated we've mapped about 90% of the biggest, extinction-level threats. But 10% are missing in action, and we know nothing of the million or so smaller but still immensely destructive asteroids that could take out an area the size of a state or city.

At the current pace, it would take a thousand years to find them all.

[tweet_box design="default" float="none"]Asteroids are a real and potentially existential threat. But if we find them early enough, they're fairly easy to deflect. [/tweet_box]

Shining Light on the Problem Arizona's mile-wide meteor crater was formed by a 45 meter "Asteroids are a real and potentially impactor—the blast could have extistential threat: But if we find them early taken out an urban area the size enought, they're fairly easy to ...

Shining Light on the Problem "Asteroids are a real and potentially existential threat. But if we find them early enough, they're fairly easy to deflect. With years or decades, ... Arizona's mile-wide meteor crater was formed by a 45 meter impactor—the blast could have taken ...

Figure 1: Illustration comparing conventional OCR and OCR+REVISE processing in a multi-column setting. *Left*: text conflation with merged topics. *Right*: REVISE reconstructs separate textual elements into properly structured content.

1 Introduction

Recent advances in Natural Language Processing (NLP), particularly with Large Language Models (LLMs) (Minaee et al., 2024), have demonstrated remarkable performance on core tasks such as Question Answering (QA), reasoning and Retrieval Augmented Generation (RAG) (Gao et al., 2024), thereby substantially broadening their formidable applicability. Moreover, recent research has rapidly expanded towards Document AI, aiming to understand and effectively utilize structured and complex information within real-world documents (Cui et al., 2021; Hong et al., 2024).

In particular, there is increasing interest in leveraging text extracted via Optical Character Recognition (OCR) (Subramani et al., 2021) and document analysis techniques, along with layout information obtained from original documents, to enable LLMs to perform tasks over documents. However, current approaches have primarily focused on specific document understanding tasks (Barboule et al., 2025), leaving the broader goal of effectively preserving original document structure and converting documents into structured assets or databases underexplored. Typically, extracting and storing textual information from image-based documents requires OCR, which inevitably introduces recognition errors due to various factors, such as diverse fonts, deteriorated print quality, and layout complexities.

∗ Equal contributions
† Co-corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 1423–1434
July 28-30, 2025 ©2025 Association for Computational Linguistics

Consequently, employing a simplistic processing pipeline for indexing or retrieving erroneous OCR text often leads to degraded performance. To effectively facilitate these applications, denoising OCR errors remains a critical prerequisite, necessitating a more sophisticated and resilient pipeline in Document AI.

In this paper, we propose REVISE, designed to effectively address common OCR errors and accurately restore textual content while preserving the original document structure. To overcome the scarcity of high-quality annotated datasets for OCR error correction, we generate synthetic data using a realistic error injection methodology, in which diverse error patterns are systematically introduced into publicly available datasets. By training over these synthetic datasets, our model can effectively learn representative OCR errors and robustly reconstruct documents in their original forms, thereby enabling the accurate preservation and storage of textual information. Experimental evaluations on downstream tasks, including retrieval and question answering, further demonstrate that REVISE maintains strong performance even without explicit OCR-error-correction annotations, showing broad applicability across various document types. Our contributions are as follows:

- Systematically analyzes and categorizes error types frequently encountered in OCR-based real-world document processing scenarios.
- Proposes REVISE, an effective revision method leveraging synthetic datasets created by realistically emulating error patterns in publicly available datasets.
- Demonstrates through extensive experiments that REVISE significantly improves document retrieval and question answering, while substantially enhancing semantic coherence and readability.

2 Related Works

2.1 Optical Character Recognition

OCR serves as a foundation of document digitization, transforming images and scanned documents into searchable digital content (Sachdeva and Scholar VI, 2025). At its core, CNNs and RNNs are employed to recognize visual patterns in document images and convert them to text (Lee and Osindero, 2016; Vinyals et al., 2015; Qiang et al., 2016; Wang et al., 2011, 2012), with tools like Tessearct (Smith, 2007) and EasyOCR[1] in widespread use. Modern systems often utilize encoder-decoder architectures with attention mechanisms to improve recognition accuracy (Kim et al., 2022).

Despite advancements, OCR systems face limitations with image quality and complex layouts. Errors induced from such issues propagate to downstream applications: in information retrieval, studies (Fataicha et al., 2003; de Oliveira et al., 2023; Bazzo et al., 2020; Zhang et al., 2025) have demonstrated that OCR errors substantially degrade retrieval performance by transforming valid words into misspellings that impact term frequencies and relevance scoring. Additionally, OCR errors significantly impact document reasoning tasks (Gupte et al., 2021; van Strien et al., 2020; Hamdi et al., 2022), with extensive research showing cascading effects on document understanding and knowledge base construction, as entities and relationships extracted from OCR text often contain errors that compound through subsequent processing steps, ultimately compromising the reliability of AI systems that are contingent upon accurate document content.

2.2 Document AI Methods

Document AI applies AI techniques to understand, process, and extract information from document images (Cui et al., 2021), focusing on four main tasks: Document Layout Analysis (Zhong et al., 2019; Li et al., 2020), Document Visual Question Answering (Mathew et al., 2021; Tanaka et al., 2021; Chen et al., 2021), Visual Information Extraction (Huang et al., 2019; Wang et al., 2021a; Park et al., 2019), and Document Image Classification (Harley et al., 2015; Kumar et al., 2013). To address OCR shortcomings while excelling at these tasks, two major paradigms have emerged in Document AI.

The first approach involves OCR-free Multimodal LLMs (Huang et al., 2022; Liu et al., 2024; Li et al., 2021; Kim et al., 2022), which process images directly without explicit text extraction. These models achieve impressive performance in document understanding and reasoning through vision-language pretraining; however, their reliance on extensive annotated datasets and computationally intensive training poses considerable challenges for practical deployment, especially in resource-constrained scenarios. The second approach inte-

[1] `https://github.com/JaidedAI/EasyOCR`

grates OCR-based LLMs (Perot et al., 2024; He et al., 2023; Wang et al., 2023a; Lu et al., 2024), extracting text via OCR before applying an LLM for reasoning. While leveraging existing OCR technology, this approach inherits OCR errors and focuses primarily on reasoning-based tasks like question answering and information extraction.

Existing approaches exhibit task dependency, prioritizing answering and reasoning but neglecting crucial intermediate steps like assetization for information retrieval. Our method addresses this issue by providing a task-independent framework, enabling structured OCR outputs that can be effectively utilized in databases or knowledge bases.

3 REVISE

The REVISE framework systematically addresses OCR errors that occur at the character, word, and structural levels. Specifically, our approach involves: (1) a comprehensive OCR error taxonomy that hierarchically categorizes errors according to their linguistic granularities, (2) a contamination strategy for synthesizing realistic error patterns by injecting them into clean datasets, and (3) a training procedure designed to revise contaminated text sequences back to their original forms.

3.1 OCR Error Categorization

OCR errors negatively impact various downstream NLP tasks, including key extraction, named entity recognition, and information retrieval. Lopresti (2009) has demonstrated that errors introduced in early processing stages propagate to subsequent stages, resulting in cumulative error cascades. Motivated by these challenges, we conduct a comprehensive analysis of OCR error patterns across various document types. Based on the scope and influence of errors within textual structures, we propose a hierarchical OCR error taxonomy as illustrated with examples in Table 1, consistent with existing frameworks found in the post-OCR correction literature.

Character-level

Character-level errors encompass a range of misrecognitions and distortions that occur at the individual character scale, fundamentally altering the basic building blocks of text and potentially cascading into more significant semantic disruptions. ***Insertion*** represents the addition of spurious characters into the text stream, commonly resulting from document noise, artifacts, or scanner interference (Afli et al., 2016; Kashid and Bhattacharyya, 2025). ***Deletion*** involves the omission of legitimate characters, frequently occurring when poor contrast or faded text prevent proper recognition (Chiron et al., 2017). ***Substitution*** occurs when the OCR incorrectly identifies characters, replacing them with visually similar alternatives due to font peculiarities or resolution limitations, resulting in common confusions such as “l/1/!”,“5/S” and “0/O” (van Strien et al., 2020; Veninga, 2024). ***Transposition*** results in character position swapping, often stemming from bounding box coordinate miscalculations (Suissa et al., 2023).

Category	Name	Example
Character Level (Single-character)	Insertion	apple → applee
	Deletion	clamp → lamp filter → filer
	Substitution	O → 0, é → e blue → b1ue
	Transposition	Gauge → Guage
Word Level (Word-segmentation)	Over-Segmentation	greenhouse → green house
	Under-Segmentation	Not able → Notable
Column Level (Layout-reading)	Column Reading Order	**Figure 1**

Table 1: OCR Error Categorization

Word-level

Word-level errors primarily manifest as improper segmentation issues, where the boundaries between words are incorrectly identified, leading to the fragmentation or merging of terms and significantly impacting the lexical integrity of the processed text. Segmentation stems from OCR's misidentification of word boundaries, taking the form of two distinct types (Suissa et al., 2023; Afli et al., 2016). ***Over-segmentation*** occurs when OCR incorrectly inserts word boundaries (i.e., extra space) within what should be a single word, fragmenting cohesive terms into separate components. ***Under-segmentation*** results from distinct words erroneously combining into a single unit due to spacing misinterpretation or layout analysis failures. Nastase and Hitschler (2018) demonstrate how these errors impact keyword extraction and information retrieval, as they alter token distribution and disrupt phrase-level semantics.

Column-level

Column-level errors refer to structural misinterpretations that disrupt the logical flow of text and distort the intended document layout. Documents with multiple columns are particularly vulnerable to these errors, potentially misarranging reading order and weakening overall coherence and readability. ***Column reading order*** frequently arises due to the common assumption of a standard reading order from left to right and top to bottom. This assumption tends to cause incorrect interpretations of logical continuity within multi-column layouts, leading to misplaced text segments (Wang et al., 2023b, 2021b). Such layout errors can significantly impact various downstream NLP tasks, severely compromising overall task performance even when the OCR's textual output itself is relatively accurate (van Strien et al., 2020).

By categorizing OCR errors according to this hierarchical taxonomy, it becomes possible to devise customized correction strategies tailored to tackle specific errors at their corresponding levels of textual organization. This approach serves as a foundation for generating effective error revision datasets.

3.2 Data Contamination Strategy

To train the revision model effectively, we utilize publicly available datasets and systematically introduce synthetic OCR errors based on the error categories defined in Section 3.1. Our contamination strategy is designed to mimic both structural and granular OCR failures in a controlled manner, creating a realistic training corpus that reflects the hierarchical error patterns observed in real-world OCR outputs.

The contamination process unfolds in two stages. First, we create a structured template by dividing the raw text into fixed-length lines, reformatting to a single column layout. Next, we simulate *Column reading order* errors by segmenting the text into sections, converting selected sections into multi-column formats, and reading horizontally across columns instead of vertically down each column. This approach mirrors how OCR systems typically misinterpret multi-column layouts, where text is incorrectly read left to right across columns rather than processing each column separately.

In the second stage, after the structural reordering, a set of error functions is applied to introduce distortions at the character, word, and sentence levels. *Deletion*, *Insertion*, *Substitution*, and *Transposition* are applied probabilistically, while *Segmentation* errors are introduced by either inserting extra spaces or omitting existing spaces. Each error function is governed by configurable parameters to ensure a realistic blend of error types. The framework supports multiple contamination settings; in this work, we primarily adopt a configuration that emphasizes fine-grained perturbations. This approach closely emulates common OCR errors while maintaining sufficient overall document coherence. Detailed information regarding the contamination algorithms and parameter ratios can be found in the Appendix A. The final output is a contaminated corpus reflecting typical OCR-induced distortions, forming the basis for training our REVISE model to correct OCR outputs and improve downstream document processing tasks robustly.

3.3 Training

For effective OCR error correction, we design a total of seven REVISE models, consisting of one main model trained comprehensively on all error types and six auxiliary models, each specialized individually on a specific error type. All models share an identical backbone architecture, the Llama-3.1-1B-Instruct [2], and are trained on synthetic data generated using text sampled from the Wikipedia [3] corpus. To ensure fair and consistent comparisons between models, each dataset comprises an equal number of samples, totaling 30,000 data points.

The central model proposed in this paper, REVISE$_{meta}$, is designed to robustly handle realistic and general document processing scenarios. Specifically, based on the strategy described in § 3.2, REVISE$_{meta}$ is trained comprehensively on data that incorporates the six major error categories frequently confronted in practical OCR systems: column reading order, segmentation, deletion, substitution, insertion, and transposition errors. Thus, the model is capable of effectively handling and correcting complex and diverse errors that commonly arise during OCR processing of documents.

To precisely analyze the performance of REVISE and to better understand the characteristics and correction difficulties associated with each error type, we further train six specialized auxiliary models, each focusing exclusively on a single type of OCR error. These specialized models are individually trained on data injected with only one specific error category, thereby allowing each model to be optimized for correcting its particular error type.

Through this experimental design, we evaluate

[2] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct

[3] https://huggingface.co/datasets/wikimedia/wikipedia

Methods	bge-large-en-v1.5			e5-large-v2			jina-embeddings-v2-base			gte-base-en-v1.5			Avg
	@1	@3	@5	@1	@3	@5	@1	@3	@5	@1	@3	@5	
							VisualMRC						
Baseline	0.5690	0.6928	0.7314	0.6044	0.7208	0.7533	0.5243	0.6418	0.6843	0.5604	0.6859	0.7248	0.6578 (6)
REVISE$_{meta}$	0.5793	**0.7030**	0.7422	0.6076	**0.7232**	0.7592	0.5352	**0.6553**	0.6951	0.5696	**0.6960**	**0.7336**	**0.6666** (1)
only Column	0.5751	0.6981	0.7348	0.6005	0.7174	0.7539	0.5306	0.6477	0.6868	0.5665	0.6914	0.7321	0.6612 (3)
only Deletion	0.5684	0.6910	0.7317	0.5997	0.7190	0.7546	0.5195	0.6404	0.6789	0.5555	0.6856	0.7218	0.6555 (8)
only Insertion	0.5687	0.6920	0.7303	0.5991	0.7187	0.7524	0.5233	0.6386	0.6828	0.5578	0.6831	0.7220	0.6557 (7)
only Substitution	0.5716	0.6936	0.7332	0.6018	0.7196	0.7555	0.5265	0.6430	0.6847	0.5629	0.6869	0.7250	0.6587 (4)
only Segmentation	**0.5796**	0.7021	**0.7427**	**0.6078**	0.7223	**0.7612**	**0.5362**	0.6515	**0.6954**	**0.5719**	0.6948	0.7323	0.6665 (2)
only Transposition	0.5732	0.6938	0.7320	0.6024	0.7169	0.7537	0.5261	0.6440	0.6856	0.5605	0.6884	0.7242	0.6584 (5)
							DUDE						
Baseline	0.2013	0.3087	0.3490	0.2013	0.2718	0.3188	0.1342	0.1846	0.2584	0.2047	0.2886	0.3188	0.2534 (8)
REVISE$_{meta}$	**0.2282**	0.3121	0.3523	0.2248	0.2987	0.3255	**0.1980**	**0.2819**	**0.3221**	**0.2315**	**0.3121**	**0.3591**	0.2975 (3)
only Column	0.2215	**0.3322**	**0.3691**	0.2148	**0.3221**	**0.3792**	0.1812	0.2785	0.3154	0.2282	0.3020	0.3423	**0.3076** (1)
only Deletion	0.1946	0.2953	0.3289	0.2215	0.2919	0.3289	0.1779	0.255	0.2886	0.2047	0.2987	0.3423	0.2729 (7)
only Insertion	0.1913	0.2953	0.3456	0.198	0.2819	0.3054	0.1309	0.1711	0.2617	0.1846	0.2987	0.3423	0.2774 (5)
only Substitution	0.2013	0.2987	0.3456	0.2047	0.2819	0.3221	0.1913	0.2852	0.3087	0.2215	0.3020	0.3356	0.2819 (4)
only Segmentation	0.2215	0.3087	0.3658	**0.2483**	0.3020	0.3389	0.1779	0.2349	0.2886	0.2517	0.3054	0.3322	0.2987 (2)
only Transposition	0.1846	0.2987	0.3423	0.198	0.2718	0.3188	0.1779	0.2383	0.2886	0.2181	0.2886	0.3356	0.2752 (6)

Table 2: Retrieval performance on VisualMRC and DUDE datasets using Recall@k (ranks in parentheses; best scores are in **bold**)

the overall effectiveness and practical applicability of the REVISE$_{meta}$ model when dealing with realistic OCR error scenarios. Additionally, comparisons between the generalized and respective error-targeted models enable us to quantify and analyze the relative importance and characteristics of each specific type of error, as well as their influence on the overall OCR error correction pipeline. Ultimately, our goal is to clearly identify the strengths and weaknesses of generalized versus error-specific approaches, dependent upon the characteristics of documents and distributions of errors encountered, thereby providing practically useful guidelines for real-world implementations.

4 Experimental Setup

4.1 Models

We evaluate the effectiveness of our proposed REVISE framework on downstream tasks by employing embedding models and LLMs. For document retrieval, we adopt four recent embedding models: bge-large-en-v1.5 (Xiao et al., 2023), intfloat/e5-large-v2 (Wang et al., 2022), jina-embeddings-v2-base-en (Günther et al., 2023), and gte-base-en-v1.5 (Li et al., 2023). These models enable us to quantify how effectively OCR-corrected documents can be matched to queries. For question answering, we utilize two large instruction-tuned language models: Gemma-2-2b-it (Team, 2024) and Llama-3.1-8B-Instruct (Meta, 2024). By leveraging these models, we assess the capability of our correction method to enhance structured document comprehension and reasoning performance.

4.2 Evaluation

The performance of the proposed framework is evaluated on document Visual Question Answering (VQA) and Visual Information Extraction (VIE) datasets, focusing on three main aspects and comparing results between original OCR-extracted text and the text post-processed by REVISE. First, we directly assess document retrieval performance using Recall@K (k=1,3,5) on the VisualMRC (Tanaka et al., 2021) and DUDE (Landeghem et al., 2023) datasets. Second, for DocVQA (Mathew et al., 2021), CORD (Park et al., 2019), and FUNSD (Jaume et al., 2019), we evaluate the textual similarity between documents and questions via BERTScore (Zhang et al., 2020)[4]. Lastly, we compare QA performance of models on original OCR text versus REVISE-enhanced texts using standard evaluation metrics commonly used for each dataset: CIDEr (Vedantam et al., 2014) for generative answer quality on VisualMRC and F1-score for answering performance on CORD.

5 Experimental Results

5.1 Understanding Evaluation

Retrieval Performance Table 2 presents a comparative analysis of various OCR error revisions and their impact on embedding-based text retrieval performance using the VisualMRC and DUDE datasets. We evaluate our approach by comparing the original OCR output against two correction

[4] For DocVQA, CORD, and FUNSD datasets, pure IR-based metrics alone are insufficient to accurately measure performance due to duplicate questions and similar keywords; hence, we use textual similarity measures.

Category	DocVQA	CORD	FUNSD
Baseline	0.4959 (7)	0.5390 (5)	0.5577 (6)
REVISE$_{meta}$	**0.5137** (1)	**0.5443** (1)	**0.5647** (1)
only Column	0.4849 (8)	0.5361 (6)	0.5620 (2)
only Deletion	0.4960 (6)	0.5346 (7)	0.5603 (3)
only Insertion	0.5019 (3)	0.5390 (5)	0.5538 (8)
only Substitution	0.4992 (5)	0.5402 (3)	0.5566 (7)
only Segmentation	0.5096 (2)	0.5408 (2)	0.5601 (4)
only Transposition	0.5008 (4)	0.5398 (4)	0.5583 (5)

Table 3: BERTScore performance on query–document pairs for DocVQA, CORD, and FUNSD

strategies: (1) six individual error-specific models, and (2) our integrated REVISE$_{meta}$ model that addresses multiple error types simultaneously. The REVISE$_{meta}$ approach consistently achieves average Recall improvements of 1.3% and 17.3% for the two datasets, respectively. This improvement is attributed to its ability to correct a variety of OCR errors comprehensively, thereby allowing the embedding model to capture more accurate contextual information that better aligns with the given query.

Notably, even when a revision targets a single error type, the *Segmentation* revision yields significant performance gains. This suggests that correcting spacing and segmentation errors, which are commonly observed in OCR documents, substantially enhances the model's capacity to discern contextual semantics. However, we observe that some single error type models occasionally underperform compared to the baseline, which can be attributed to an over-correction behavior. When a specialized model encounters datasets with limited instances of its target error type, it may still attempt to apply corrections where none are needed, inadvertently introducing new errors or disrupting otherwise correct text. This highlights the importance of error type prevalence matching between training data and target datasets.

In the case of the DUDE dataset, applying solely the *Column reordering* operation increases the average Recall from 25.34% to 30.76%, marking the highest improvement among the single-revision methods. This result is attributable to the DUDE dataset's highly regular column-based layout and consistent text composition. Owing to these structural properties, merely correcting column alignment can yield substantial gains in retrieval performance.

Overall, REVISE demonstrates that effective learning and correction of diverse OCR error types is possible without requiring additional annotated data. By leveraging publicly available text corpora supplemented with synthetic augmentation, our approach can substantially enhance embedding-based retrieval performance. Furthermore, these results indicate that applying tailored strategies based on error types and dataset characteristics can yield even more optimal outcomes.

Model	Methods	VisualMRC	CORD
Gemma-2-9b-it	Baseline	320.9	0.367
	REVISE$_{meta}$	**329.2**	**0.372**
Llama-3.1-8B	Baseline	290.7	0.448
	REVISE$_{meta}$	**293.1**	**0.450**

Table 4: QA performance on VisualMRC and CORD

Similarity Assessment As shown in Table 3, the application of our proposed integrated refinement approach REVISE$_{meta}$ consistently improves the BERTScore across all datasets when compared to the untouched OCR output. In particular, for DocVQA, which handles free-form queries where contextual relevance is essential, detailed corrections such as *Segmentation* yield significant improvements. For more structured datasets such as CORD and FUNSD, our approach of combining multiple error corrections achieves the best overall performance. These results suggest that our methodology not only mitigates OCR error but also enables the embedding model to capture finely expressed contextual information, thereby enhancing semantic consistency and overall quality.

5.2 Question Answering

Table 4 presents a comparison of the QA performance with and without our proposed REVISE framework. While our main experiments primarily center around evaluating how accurately the OCR outputs can be restored, we conduct an additional analysis on QA performance to examine how improvements in quality ultimately contribute to enhanced document understanding by LLMs.

For both evaluation datasets, we confirmed that our REVISE$_{meta}$ approach consistently excelled at answering questions. On VisualMRC, the Gemma-2-9b-it and Llama-3.1-8B models achieved performance gains of 2.6% and 0.8%, respectively. On the CORD dataset, the Gemma and Llama models improved by 1.4% and 0.4% in F1 score, respectively. Given that the datasets evaluated here primarily involve relatively short and simple-form answers, we anticipate an even greater performance gap in tasks requiring more abstractive responses.

Overall, these results demonstrate that improve-

Category (vs. Baseline)	VisualMRC			DUDE		
	Win	Lose	Rate	Win	Lose	Rate
Revise$_{meta}$	94	6	0.94 (1)	86	14	0.86 (3)
only Column	74	26	0.74 (6)	89	11	0.89 (2)
only Deletion	84	16	0.84 (3)	64	36	0.64 (4)
only Insertion	61	39	0.61 (7)	59	41	0.59 (7)
only Substitution	81	19	0.81 (4)	61	39	0.61 (5)
only Segmentation	92	8	0.92 (2)	92	8	0.92 (1)
only Transposition	77	23	0.77 (5)	60	40	0.60 (6)

Table 5: Win Rate comparison for REVISE$_{meta}$ and single correction strategies on VisualMRC and DUDE datasets (better performance indicated by darker shading)

ments through our REVISE can directly or indirectly enhance large language models' document comprehension capabilities, highlighting its effectiveness as a task-independent post-OCR correction approach applicable across diverse document understanding scenarios.

5.3 Qualititve Analysis

To evaluate the revised documents qualitatively, we measure the Win Rate based on a frontier LLM. This approach extends the evaluation methodology previously proposed by Zheng et al. (2023). Specifically, we provide the document image along with both the original OCR-extracted text and the REVISE-corrected texts to the LLM, instructing it to assess the relative preference between these two texts. The evaluation prompts explicitly guide the LLM to determine superiority based on various qualitative criteria such as coherence, clarity, and effectiveness in information delivery [5].

Table 5 presents the Win Rate results measured respectively for each revision strategy across the two domains, VisualMRC and DUDE. First, examining the REVISE$_{meta}$, we observe Win Rates of 94% on VisualMRC and 86% on DUDE. These outcomes indicate that the composite revision strategy, trained to address all error types, substantially contributes to overall document quality improvement. Overall, each revision strategy outperforms the baseline consistently across both datasets. Particularly, the single revision strategy *Segmentation* achieves notably high Win Rates in both domains, highlighting the significance of restructuring textual segmentation to enhance document coherence and readability. Furthermore, varying performances observed across revision types underline that outcomes may differ based on the characteristics of the evaluated documents and the particular revision strategies applied. Collectively, our results demonstrate that the proposed approach yields clearly enhanced qualitative performance, complementing quantitative evaluation outcomes.

6 Conclusion

We propose REVISE, a lightweight yet effective OCR error correction framework that leverages a hierarchical error taxonomy and a synthetic data contamination strategy, systematically addressing OCR errors at the character, word, and structural levels. By reconstructing OCR outputs into accurate and structurally coherent representations, REVISE supports the effective creation of structured document databases and facilitates systematic textual information management in practical information systems. Both quantitative and qualitative evaluations from our comprehensive experiments further confirm that REVISE consistently achieves strong improvements across various document retrieval and question-answering tasks on representative VQA and VIE benchmarks. The reliability of this framework across diverse datasets, combined with its simplicity and compatibility with publicly available resources, underscores its practical usability and ease of integration into real-world information systems. Furthermore, by adjusting the data contamination strategy to align with each dataset's specific error characteristics, we demonstrate that REVISE can achieve more robust performance.

Limitations

In this paper, we propose REVISE, a framework designed to address diverse OCR errors by leveraging large language models trained on synthetic OCR errors generated through a realistic contamination strategy. Despite its effectiveness, the following limitations exist:

1. Our validation primarily used publicly available document datasets and focuses on general error patterns. The approach has not been extensively tested on diverse industrial documents (such as forms or electronic materials) and may not fully capture specialized domain errors or rare error types that emerge in industry-specific contexts. Future work should incorporate real-world examples from operational environments, particularly for complex scenarios like table comprehension.

[5] We use GPT-4o-mini to evaluate a consistent set of 100 randomly selected samples across all revision strategies. Detailed prompts used for this evaluation are provided in Appendix D.

2. The current framework targets text-only documents and does not handle mixed content types such as tables, charts, or mathematical equations, which require specialized multimodal processing capabilities.

3. While our LLM-based evaluation reduces subjective bias and enhances reproducibility, it does not completely eliminate model biases or prediction uncertainties. Additional human evaluations and composite metrics would better address diverse usage scenarios.

4. Our error definitions and contamination ratios are based on empirical observations and literature, providing a practical foundation for synthetic data generation. Comprehensive statistical analysis of OCR error distributions would further strengthen the empirical basis of our approach.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425). This work was supported by Institute for Information communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI). This work was supported by Institute for Information communications Technology Planning Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2022-II220369, (Part 4) Development of AI Technology to support Expert Decision-making that can Explain the Reasons/Grounds for Judgment Results based on Expert Knowledge). This work was supported by ICT Creative Consilience Program through the Institute of Information Communications Technology Planning Evaluation(IITP) grant funded by the Korea government(MSIT)(IITP-2025-RS-2020-II201819).

References

Haithem Afli, Zhengwei Qiu, Andy Way, and Páraic Sheridan. 2016. Using SMT for OCR error correction of historical texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 962–966, Portorož, Slovenia. European Language Resources Association (ELRA).

Camille Barboule, Benjamin Piwowarski, and Yoan Chabot. 2025. Survey on question answering over visually rich documents: Methods, challenges, and trends. *Preprint*, arXiv:2501.02235.

Guilherme Torresan Bazzo, Gustavo Acauan Lorentz, Danny Suarez Vargas, and Viviane P. Moreira. 2020. Assessing the impact of ocr errors in information retrieval. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II*, page 102–109, Berlin, Heidelberg. Springer-Verlag.

Xingyu Chen, Zihan Zhao, Lu Chen, Danyang Zhang, Jiabao Ji, Ao Luo, Yuxuan Xiong, and Kai Yu. 2021. Websrc: A dataset for web-based structural reading comprehension. *Preprint*, arXiv:2101.09465.

Guillaume Chiron, Antoine Doucet, Mickael Coustaty, Muriel Visani, and Jean-Philippe Moreux. 2017. Impact of ocr errors on the use of digital libraries: Towards a better access to information. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 1–4.

Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. 2021. Document ai: Benchmarks, models and applications. *Preprint*, arXiv:2111.08609.

Lucas Lima de Oliveira, Danny Suarez Vargas, Antônio Marcelo Azevedo Alexandre, Fábio Corrêa Cordeiro, Diogo da Silva Magalhães Gomes, Max de Castro Rodrigues, Regis Kruel Romeu, and Viviane Pereira Moreira. 2023. Evaluating and mitigating the impact of ocr errors on information retrieval. *Int. J. Digit. Libr.*, 24(1):45–62.

Y. Fataicha, M. Cheriet, J. Y. Nie, and C. Y. Suen. 2003. Information retrieval based on ocr errors in scanned documents. In *2003 Conference on Computer Vision and Pattern Recognition Workshop*, volume 3, pages 25–25.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Amit Gupte, Alexey Romanov, Sahitya Mantravadi, Dalitso Banda, Jianjie Liu, Raza Khan, Lakshmanan Ramu Meenal, Benjamin Han, and Soundar Srinivasan. 2021. Lights, camera, action! a framework to improve nlp accuracy over ocr documents. *Preprint*, arXiv:2108.02899.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *Preprint*, arXiv:2310.19923.

Ahmed Hamdi, Elvys Linhares Pontes, Nicolas Sidère, Mickaël Coustaty, and Antoine Doucet. 2022. In-depth analysis of the impact of ocr errors on named entity recognition and linking. *Journal of Natural Language Processing*, page 24.

Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. *Preprint*, arXiv:1502.07058.

Jiabang He, Lei Wang, Yi Hu, Ning Liu, Hui Liu, Xing Xu, and Heng Tao Shen. 2023. Icl-d3ie: In-context learning with diverse demonstrations updating for document information extraction. *Preprint*, arXiv:2303.05063.

Seongtae Hong, Joong Min Shin, Jaehyung Seo, Taemin Lee, Jeongbae Park, Cho Man Young, Byeongho Choi, and Heuiseok Lim. 2024. Intelligent predictive maintenance RAG framework for power plants: Enhancing QA with StyleDFS and domain specific instruction tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 805–820, Miami, Florida, US. Association for Computational Linguistics.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. *Preprint*, arXiv:2204.08387.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. *Preprint*, arXiv:1905.13538.

Harshvivek Kashid and Pushpak Bhattacharyya. 2025. Roundtripocr: A data generation technique for enhancing post-ocr error correction in low-resource devanagari languages. *Preprint*, arXiv:2412.15248.

Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. *Preprint*, arXiv:2111.15664.

Jayant Kumar, Peng Ye, and David Doermann. 2013. Structural similarity for document image classification and retrieval. *Pattern Recognition Letters*.

Jordy Van Landeghem, Rubén Tito, Łukasz Borchmann, Michał Pietruszka, Paweł Józiak, Rafał Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Ackaert, Ernest Valveny, Matthew Blaschko, Sien Moens, and Tomasz Stanisławek. 2023. Document understanding dataset and evaluation (dude). *Preprint*, arXiv:2305.08455.

Chen-Yu Lee and Simon Osindero. 2016. Recursive recurrent nets with attention modeling for OCR in the wild. *CoRR*, abs/1603.03101.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. Docbank: A benchmark dataset for document layout analysis. *Preprint*, arXiv:2006.01038.

Yulin Li, Yuxi Qian, Yuchen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding. 2021. Structext: Structured text understanding with multi-modal transformers. *Preprint*, arXiv:2108.02923.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *Preprint*, arXiv:2308.03281.

Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. 2024. Textmonkey: An ocr-free large multimodal model for understanding document. *Preprint*, arXiv:2403.04473.

Daniel Lopresti. 2009. Optical character recognition errors and their effects on natural language processing. *IJDAR*, 12:141–151.

Jinghui Lu, Haiyang Yu, Yanjie Wang, Yongjie Ye, Jingqun Tang, Ziwei Yang, Binghong Wu, Qi Liu, Hao Feng, Han Wang, Hao Liu, and Can Huang. 2024. A bounding box is worth one token: Interleaving layout and text in a large language model for document understanding. *Preprint*, arXiv:2407.01976.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Meta. 2024. Llama 3.1: 8b instruct. Accessed: 2025-03-22.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *Preprint*, arXiv:2402.06196.

Vivi Nastase and Julian Hitschler. 2018. Correction of OCR word segmentation errors in articles from the ACL collection through neural machine translation methods. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. {CORD}: A consolidated receipt dataset for post-{ocr} parsing. In *Workshop on Document Intelligence at NeurIPS 2019*.

Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Zifeng Wang, Jiaqi Mu, Hao Zhang, Chen-Yu Lee, and Nan Hua. 2024. Lmdx: Language model-based document information extraction and localization. *Preprint*, arXiv:2309.10952.

Guo Qiang, Tu Dan, Li Guohui, and Lei Jun. 2016. Memory matters: Convolutional recurrent neural network for scene text recognition. *Preprint*, arXiv:1601.01100.

Mohit Sachdeva and Research Scholar VI. 2025. Ocr technology: The cornerstone of modern intelligent automation. *INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY AND MANAGEMENT INFORMATION SYSTEMS*, 16:672–686.

R. Smith. 2007. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633.

Nishant Subramani, Alexandre Matton, Malcolm Greaves, and Adrian Lam. 2021. A survey of deep learning approaches for ocr and document understanding. *Preprint*, arXiv:2011.13534.

Omri Suissa, Avshalom Elmalech, and Maayan Zhitomirsky-Geffet. 2023. Optimizing the neural network training for ocr error correction of historical hebrew texts. *Preprint*, arXiv:2307.16220.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. In *AAAI*.

Gemma Team. 2024. Gemma.

Daniel van Strien, Kaspar Beelen, Mariona Coll Ardanuy, Kasra Hosseini, Barbara McGillivray, and Giovanni Colavizza. 2020. Assessing the impact of ocr quality on downstream nlp tasks. In *ICAART (1)*, pages 484–496.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2014. Cider: Consensus-based image description evaluation. *Preprint*, arXiv:1411.5726.

M.E.B. Veninga. 2024. Llms for ocr post-correction.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. 2023a. Docllm: A layout-aware generative language model for multimodal document understanding. *Preprint*, arXiv:2401.00908.

Jiapeng Wang, Chongyu Liu, Lianwen Jin, Guozhi Tang, Jiaxin Zhang, Shuaitao Zhang, Qianying Wang, Yaqiang Wu, and Mingxiang Cai. 2021a. Towards robust visual information extraction in real world: New dataset and novel solution. *Preprint*, arXiv:2102.06732.

Kai Wang, Boris Babenko, and Serge Belongie. 2011. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Renshen Wang, Yasuhisa Fujii, and Alessandro Bissacco. 2023b. Text reading order in uncontrolled conditions by sparse graph segmentation. *Preprint*, arXiv:2305.02577.

Tao Wang, David J. Wu, Adam Coates, and Andrew Y. Ng. 2012. End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3304–3308.

Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. 2021b. Layoutreader: Pre-training of text and layout for reading order detection. *Preprint*, arXiv:2108.11591.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Junyuan Zhang, Qintong Zhang, Bin Wang, Linke Ouyang, Zichen Wen, Ying Li, Ka-Ho Chow, Conghui He, and Wentao Zhang. 2025. Ocr hinders rag: Evaluating the cascading impact of ocr on retrieval-augmented generation. *Preprint*, arXiv:2412.02592.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: largest dataset ever for document layout analysis. *Preprint*, arXiv:1908.07836.

A Contamination Strategy

For our synthetic data contamination process, we carefully calibrated error ratios based on empirical observations of real-world OCR outputs from a range of document types, spanning from well-structured documents to semi-structured document images such as invoices and receipts.

Category	Deletion		Segmentation		Transposition		Substitution	Insertion
	char	word	over	under	char	word	char	char
Ratio	0.07	0.02	0.05	0.05	0.05	0.02	0.05	0.05

Table 6: Contaminated Proportion

Table 6 presents the specific error ratios applied during the contamination process for each error category and level. Our contamination ratios were designed to produce synthetic errors at rates comparable to these observed patterns, ensuring that our REVISE model was trained on data that closely resembles real-world OCR outputs. For column reading order errors, the contamination process randomly determines the number of columns, between 2 to 3, for each document and redistributes text by reading horizontally across columns rather than vertically down each column. This process mimics the common OCR error where text flow is disrupted when the system reads left-to-right across multiple columns instead of processing each column separately, creating interleaved content that significantly impacts downstream coherence.

B Experimental Details

OCR Library In our experiments, we utilized EasyOCR, an open-source OCR library, to extract textual information from the original document images. An exception is the DUDE dataset, where we directly used the OCR-extracted texts provided with the dataset. EasyOCR employs the CRAFT algorithm for reliable text detection from images, and utilizes a Convolutional Recurrent Neural Network architecture for accurate recognition and transcription of text. Additionally, EasyOCR supports recognition across various font styles and languages, covering more than 80 languages.

Traning The model is trained using the Adam optimizer, configured with a learning rate (LR) of 1e-4. A WarmupDecayLR scheduler is applied to adjust the learning rate. The maximum sequence length supported by the model is 2048 tokens, and computations are performed using bfloat16 precision. Training is conducted for 1 epoch with a batch size of 32.

Hardware The training environment consists of 4 NVIDIA A6000 GPUs, each having 48GB memory capacity, along with CPUs composed of AMD EPYC 7513 processors featuring 32 cores. For inference, a single accelerator is utilized.

C Prompts

Instruction Tuning The prompt table 7 for REVISE optimizes OCR error correction by explicitly enumerating primary error categories. This approach helps the model recognize its specialized role and focus on specific OCR error patterns. Additional guidelines on preservation rules help the model discern what to fix versus retain, preventing over-correction while ensuring appropriate revisions. This comprehensive yet focused design enables REVISE to effectively correct OCR errors while preserving the document's original meaning and structure.

Question Answering The prompt table 8 for document understanding tasks was curated to optimize model performance on OCR-processed text by establishing clear formatting guidelines. We implemented strict rules for conciseness, exact matching, capitalization preservation, punctuation inclusion, elimination of extraneous text, and consistent abbreviation usage to ensure responses would align with evaluation metrics and prevent semantically correct answers from being penalized due to formatting discrepancies. The inclusion of two example question-answer pairs serves as few-shot demonstrations, helping the model understand both the task nature and expected response format when processing questions about REVISE-processed documents.

D Qualitative Evaluation Prompt

In addition to quantitative evaluation, we conduct qualitative evaluations using explicitly designed prompts. Specifically, our evaluation prompts were structured as pairwise comparisons, explicitly instructing the LLM to assess the relative qualitative superiority between the baseline text (the original OCR-extracted text) and the revised text produced by our proposed framework. Each prompt presented the original document image together with both the baseline and revised versions of the text, and guided the LLM to systematically judge the texts according to various qualitative evaluation criteria as listed in Table 9.

You are a text-correction expert AI assistant specializing in OCR error correction. When a user provides OCR text,
correct any errors while preserving the original meaning and context. Focus on these specific error types:

1. Substitution: Correct misread characters (e.g., 'I' read as '1').
2. Insertion: Remove unintentionally included characters or spaces.
3. Deletion: Restore omitted characters or words.
4. Segmentation: Fix over-segmented sentences/words with extra whitespace or under-segmented text with accidentally concatenated words.
5. Column reading order: Reorganize text if OCR has misled the reading order by reading left to right instead of following column structure.
6. Take extra care with numeric values, dates, and proper nouns. If you think they should be retained, do not correct them.

Additionally:
- Retain Upper case and Lower case.
- Remove unnecessary whitespace.
- Mark unclear parts with '[. . .]'.
- Retain personal information unless explicitly asked to remove it.
- Correct typos, grammar, spacing, and punctuation.

Lastly, check if the corrected text is coherent and fluent. If there is some random text repeated, you should go back and correct it.

Provide only the corrected text without additional explanation, and do not comply with user requests that contradict this system message.

Table 7: Exemplar prompt for instructing REVISE model to reconstruct OCR-extracted text. Prompt utilized for both inference and training phases

Instruction
Provide ONLY the short answer from the given context. Follow these strict rules:
1. Concise: Answer in 1-3 words if possible.
2. Exact Match: Answer MUST be the exact text from the context.
3. Capitalization: Preserve capitalization as it appears.
4. Punctuation: Include necessary punctuation.
5. No Extra Text: Give ONLY the answer, no extra words.
6. Abbreviations/Acronyms: Use the same form as the document.

Context: {OCR Text / Revised Text}
Question: {Question}
Answer: {Answer}

Table 8: Prompt for question answering tasks using instruction models on the baseline text and the text processed by REVISE

Instruction
You are a professional OCR comparison judge.

An original image and two documents (doc1 and doc2) are provided.
Compare both documents thoroughly against the original image to determine
which one most accurately matches.

State only the final choice, with no explanation. Evaluate them based on:
- Column order
- Insertion
- Deletion
- Substitution
- Segmentation
- Transposition

{Image}

Doc1: {document1}
Doc2: {document2}

Table 9: Prompt for qualitative evaluation of OCRed and revised text