StruNRAG: Evaluation of OCR-Induced Structural Noise on RAG Robustness

Mengna Gao[1,2], **Dapeng Yin**[1,2], **Shuyue Zhu**[1,2]
Bingxuan Hou[1,2], **Zhanpeng Ni**[1,2], **Junli Wang**[1,2*]
[1] Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, Shanghai 201804, China.
[2] National (Province-Ministry Joint) Collaborative Innovation Center for Financial Network Security, Tongji University, Shanghai 201804, China.
{2432121, 2432122, 2432272, 2432023, 2534192, junliwang}@tongji.edu.cn

Abstract

Retrieval-Augmented Generation (RAG) systems rely on Optical Character Recognition (OCR) to ingest knowledge from unstructured documents. However, OCR engines often struggle with complex layouts, introducing **Structural Noise**, such as line insertion and paragraph interleaving, which disrupts the semantic flow of the text. Existing evaluations largely overlook this dimension, operating on the assumption of structurally perfect input. To bridge this gap, we introduce StruNRAG, a dedicated benchmark for evaluating RAG robustness against OCR-induced structural perturbations. We construct a bilingual dataset of 2,132 question-answer pairs derived from complex Chinese and English documents and systematically inject three categories of real-world structural noise: line insertion, paragraph interleaving, and line interleaving. Our evaluation of mainstream retrievers and Large Language Models (LLMs) reveals a nuanced interaction between noise and pipeline stages: while structural distortions consistently degrade retrieval performance, the generation stage exhibits unexpected robustness. Advanced LLMs demonstrate robustness against local noise (e.g., line insertion), but struggle to maintain reasoning capabilities under severe structural disruption that fragments global context. These findings indicate that while LLMs are capable of compensating for minor parsing errors, future RAG optimizations must take into account the effects of structural noise. Our code and datasets are available at https://github.com/GaoMengnana/StruNRAG.

1 Introduction

Retrieval-Augmented Generation (RAG) supplements the knowledge base of Large Language Models (LLMs) by retrieving external documents (Lewis et al., 2020), enabling them to address queries beyond their pre-training scope while mitigating model hallucinations (Izacard et al., 2023; Li et al., 2023). In the real world, a vast amount of information exists in unstructured formats, with PDF documents being particularly prevalent; due to their diverse content and layouts, utilizing Optical Character Recognition (OCR) technology for document parsing has become a critical link in the construction of RAG knowledge bases (Wang et al., 2024a). However, evaluations indicate that for documents with complex layouts (Xu et al., 2020, 2021), the precision of OCR often reaches only 30 % - 50% (Ouyang et al., 2025); furthermore, OHR-Bench (Zhang et al., 2025) notices that knowledge bases constructed from OCR results exhibit a performance gap of at least 14% compared with versions based on ground-truth (GT) data.

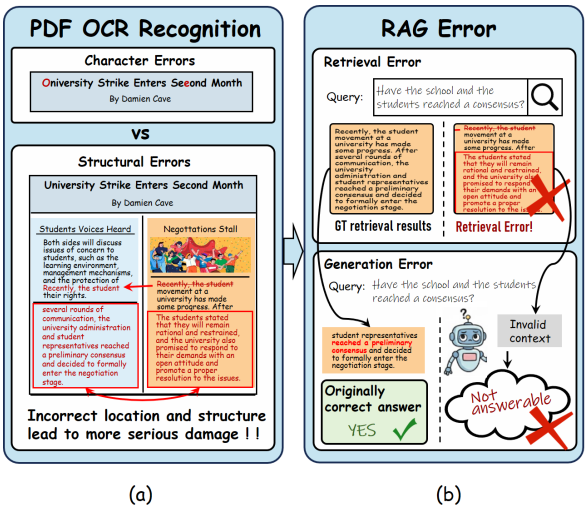

Figure 1: Complex layouts induce OCR parsing failures, introducing structural noise that degrades retrieval quality and leads to erroneous generation results.

Although OHR-Bench investigates the noise arising from the knowledge base construction process, its analysis remains limited to the character

*Corresponding author. This work was supported by the National Key Research and Development Program of China under Grant 2023YFB3002201.

Findings of the Association for Computational Linguistics: ACL 2026, pages 19129–19148

level. Commonly used open-source parsing tools (Li, 2024; Varma et al., 2023) and OCR models (Liao et al., 2023; Fang et al., 2025; Zhang et al., 2024) often fail to faithfully reconstruct the structural information (e.g., paragraph boundaries, section orders) in complex documents (Ouyang et al., 2025). This limitation triggers a more fundamental and destructive phenomenon, which we refer to as **Structural Noise**. As illustrated in Figure 1 (a), unlike traditional character-level perturbations, structural noise disrupts the spatial-logical coherence of documents while preserving literal content. We categorize such distortions into Line Insertion, Paragraph Interleaving, and Line Interleaving. This noise propagates through the RAG pipeline causing a cascading failure: as shown in Figure 1 (b), it induces semantic fragmentation that misguides retrieval, and destroys the contextual integrity essential for generation, ultimately leading to factual errors or response refusals.

Existing research on RAG system evaluation (Es et al., 2024; Saad-Falcon et al., 2024) and noise robustness mostly focuses on internal processes such as retrieval noise (Karpukhin et al., 2020; Izacard et al., 2023), while there is insufficient discussion on the noise inherent in the knowledge base (Shen et al., 2021; Pfitzmann et al., 2022); consequently, the performance of RAG systems in the presence of structural noise remains unknown, lacking both a targeted benchmark and in-depth attribution analysis.

Therefore, building upon this consideration and conceptual framework of **Stru**ctural **N**oise, we develop the **StruNRAG** Benchmark; by simulating typical structural defects in documents, we construct a dataset comprising 2,132 question-answering pairs and a retrieval knowledge base containing three types of structural noise. Based on this benchmark, we evaluate several mainstream retrievers and generative models. Experimental results demonstrate that structural noise leads to a "cascading performance decay" in RAG systems; although LLMs exhibit some robustness during generation, end-to-end performance still suffers a precipitous drop under severe structural perturbations. This reveals an under-recognized source of vulnerability within RAG systems, providing a theoretical foundation and empirical evidence for the future development of more robust OCR techniques and anti-interference RAG architectures. The main work and contributions of this paper are as follows:

- This paper explicitly defines three typical categories of OCR structural noise, including line insertion, paragraph interleaving, and line interleaving.
- By simulating various types of structural noise on PDF documents with complex layouts, this study constructs a RAG-oriented question-answering benchmark to systematically analyze the impact of structural noise on information retrieval and generation quality.
- Through a systematic experimental evaluation framework, this paper quantifies the influence of different noise categories on RAG performance across the retrieval, generation, and end-to-end stages.

2 Related Work

2.1 Document Parsing and Structural Recognition

The evolution of OCR has shifted from basic character recognition to the structural understanding of complex documents (Li, 2024; Varma et al., 2023; Wang et al., 2024a). Pipeline-based systems (Zhou et al., 2017; Liao et al., 2023; Shi et al., 2017) decompose OCR into modular components. However, cascading models often fragment context, making pipelines prone to structural errors in reading order and element association (Pfitzmann et al., 2022; Shen et al., 2021). End-to-end approaches (Blecher et al., 2023; Fang et al., 2025) jointly optimize sub-tasks for coherent output. Still, lacking fine-grained Document Layout Analysis (DLA), these end-to-end approaches struggle to preserve spatial document relationships, resulting in practical structural misalignment (Wang et al., 2024b). LVLMs (Zhang et al., 2024; Ye et al., 2023) extend cross-modal reasoning, but their CLIP-based visual encoders (Radford et al., 2021) emphasize coarse alignment, insufficiently capturing detailed layout structures for OCR. Despite continuous OCR progress, structured parsing of complex PDFs inevitably introduces structural recognition errors.

2.2 RAG Evaluation

Evaluation frameworks are critical for diagnosing the performance of RAG pipelines. Standard benchmarks such as RAGAS and ARES (Es et al., 2024; Saad-Falcon et al., 2024) evaluate

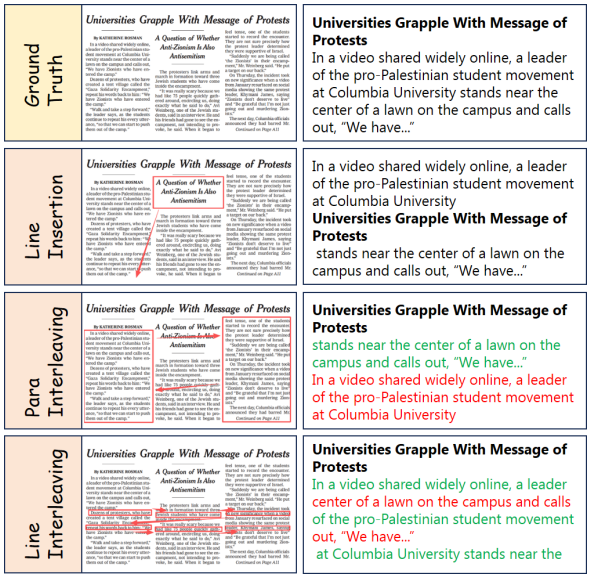

Figure 2: Examples of structural noise introduced during OCR-based PDF parsing. The colored areas represent reversed positions.

RAG systems across dimensions like context relevance, faithfulness, and answer relevance. RAG-QA (Han et al., 2024) and Adaptive-RAG (Jeong et al., 2024) assess model robustness against general noise and negative samples, while MultiHop-RAG (Lv et al., 2021) focuses on the ability to perform complex reasoning across multiple documents. Despite these advances, existing methods provide limited discussion on external knowledge base quality. Current RAG evaluation typically treats OCR-parsed text as trustworthy input (Shen et al., 2021; Pfitzmann et al., 2022), failing to capture structural issues like reading order confusion introduced during parsing. Regarding OCR noise analysis, OHR-Bench (Zhang et al., 2025) categorizes errors into format and semantic noise, but predominantly concentrates on character-level detection, there remains a lack of research quantifying the impact of structural noise on RAG performance.

3 OCR Structural Noise

This section provides motivation and a formal description of structural noise introduced by OCR systems.

3.1 Motivation

We conduct a statistical study on 109 journal PDF documents with intricate layouts, collected following the procedure described in DocBench (Zou et al., 2025). We employ MinerU [1] to parse these documents and analyze the resulting outputs. To intuitively evaluate the text parsing quality of the MinerU method, we statistically analyzed the character error rate (CER) of the parsing results and found that the CER reaches 75.37%, which indicates serious quality issues. In addition, we manually count the occurrences of structural noise in the OCR outputs and observe an average of 3.9 structural noise instances per document. Detailed statistics are reported in Appendix A.

These results highlight the prevalence and severity of structural noise in OCR outputs for complex-layout documents, underscoring the necessity of systematically investigating and evaluating its impact.

3.2 Definition of Three Types of Structural Noise

In this paper, **Structural Noise** refers to the systematic disruption of semantic coherence when transforming raw document layouts into linearized text streams. This disruption typically arises from misjudgments by layout analysis algorithms regarding logical reading order, physical boundaries, or hierarchical relationships.

Given the inherent challenges of OCR in complex layouts, we identify and formalize three primary categories of structural noise. Figure 2 provides illustrative examples for each category.

Line Insertion (LInsert) Occurs when DLA incorrectly embeds semantically isolated elements (e.g., headers) into coherent chunks $[l_1, \ldots, l_n]$. This yields a corrupted sequence $[l_1, \ldots, l_i, l_{noise}, l_{i+1}, \ldots, l_n]$, this sequence is defined as LInsert noise, where l_{noise} is irrelevant inserted line.

Paragraph Interleaving (PInterl) Stemming from misinterpreted logical flows, this noise transforms adjacent paragraphs p_A and p_B into an alternating structure $p_{noise} = [p_{A_1}, p_{B_1}, p_{A_2}, p_{B_2}]$, which is defined as PInterl noise, and p_{A_i} denotes fragmented sub-segment of p_A.

Line Interleaving (LInterl) Arises from failed multi-column boundary detection, where lines from distinct chunks $L_{left} = [l^1_{left}, l^2_{left}, \ldots]$ and $L_{right} = [l^1_{right}, l^2_{right}, \ldots]$ are intermixed by vertical coordinates, resulting in the parsed output

[1] https://github.com/opendatalab/MinerU/releases/tag/magic_pdf-1.3.2-released

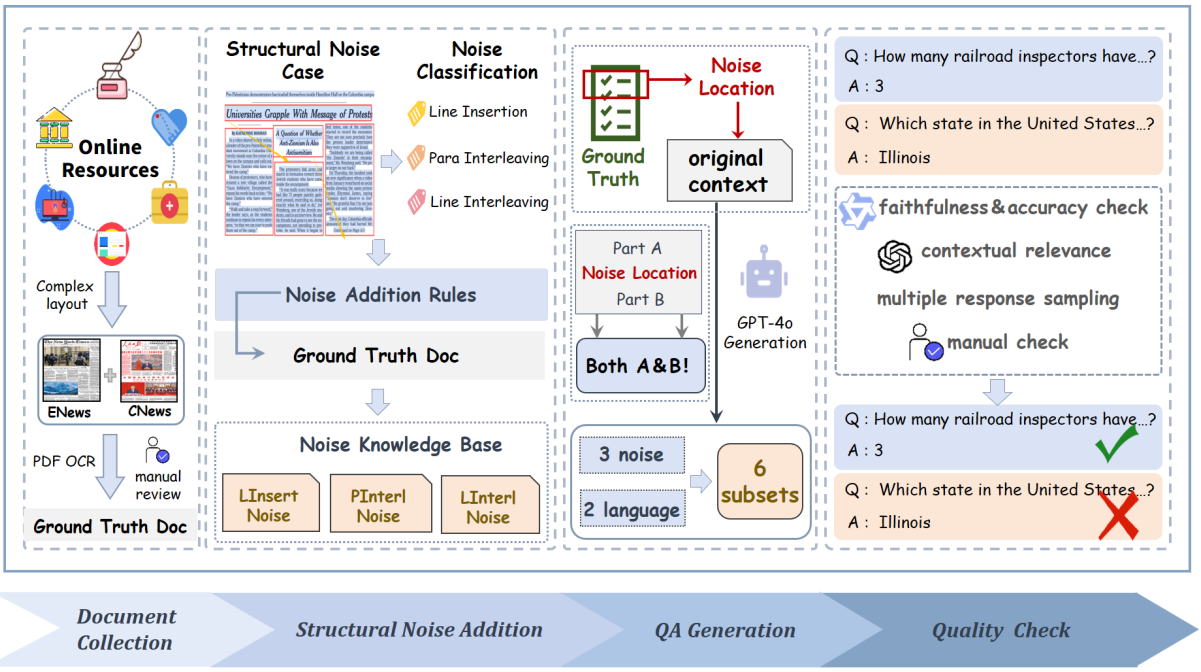

Figure 3: Overview of the StruNRAG benchmark construction pipeline. (1) Sourcing bilingual news documents with complex layouts from online repositories. (2) Systematically injecting three categories of structural noise into the GT text. (3) Synthesizing question-answering pairs based on the GT context. (4) Employing a multi-stage verification process to filter and refine high-quality QA pairs.

$li_{noise} = [l^1_{left}, l^1_{right}, l^2_{left}, l^2_{right}, \dots]$, this output is defined as LInterl noise, where l^i_{left} denotes i^{th} line of the left chunk.

4 StruNRAG Benchmark

We introduce StruNRAG, a benchmark designed to evaluate RAG robustness against structural noise in complex layouts. It comprises 2,132 bilingual questions across six subsets and a noisy knowledge base derived from complex PDF parsing. Figure 3 illustrates the construction pipeline.

4.1 Data Collection

Structural noise often originates from complex document layouts, and newspaper articles are particularly prone to such issues due to multi-column formatting, mixed text-image content, and irregular block structures (Xu et al., 2021, 2020). Therefore, we collect PDF documents from two widely used news sources: English *The New York Times* (NYT) and Chinese *Peoples Daily* (PD). In total, the dataset consists of 425 PDF documents.

We adopt a hybrid annotation pipeline that combines automated processing with manual verification to construct GT. Specifically, we employ the Windows.media.ocr API[2] to perform initial text recognition, then, professional annotators conduct thorough manual review and correction, ensuring that the final text sequence faithfully reflects the original PDF layout while minimizing structural biases introduced by automated tools. Human evaluation criteria is detailed in Appendix C.1.

4.2 Addition of Structural Noise

To quantitatively evaluate the individual impacts of the three types of structural noise on RAG systems, we inject these noise categories into the GT knowledge base according to predefined noise-addition rules. From a unified perspective, we model structural noise as a document-level transformation operator. Formally, let x denote a GT document designated for noise injection. A noisy document $\tilde{x}$ is generated by applying a structural noise operator:

$$\tilde{x} = \mathcal{N}(x), \tag{1}$$

where $\mathcal{N}(\cdot)$ denotes the noise injection process. Each noise type corresponds to a specific instantiation of $\mathcal{N}$ operating at different structural levels.

LInsert Noise. Based on the definitions established in Section 3.2, the process of introducing LInsert noise can be formally defined as follows:

[2] https://learn.microsoft.com/en-us/uwp/api/windows.media.ocr

		Chinese(PD)			English(NYT)		
	MinerU (Wang et al., 2024a)	LInsert	PInterl	LInterl	LInsert	PInterl	LInterl
CER(%)	75.37	11.09	59.45	57.00	12.63	51.56	51.52

Table 1: Impact of specific structural noise categories on Chinese and English datasets.

$$L_{\text{noise}} = \{l^1_{noise}, l^2_{noise}, \ldots, l^m_{noise}\}, \quad \tilde{x}_{\text{LInsert}} = \text{Insert}(x, L_{noise}), \tag{2}$$

where L_{noise} denotes the set of candidate noise rows to be inserted; m represents the number of noise insertion operations; Insert$(\cdot)$ signifies the operation of adding noise rows into the document; and $\tilde{x}_{\text{LInsert}}$ expresses the final output noisy document.

PInterl Noise. Based on the definitions established in Section 3.2, the process of introducing PInterl noise can be formally defined as follows:

$$P_{\text{noise}} = \{p^1_{noise}, p^2_{noise}, \ldots, p^m_{noise}\}, \quad \tilde{x}_{\text{PInterl}} = \text{PInterleave}(x, P_{noise}), \tag{3}$$

where P_{noise} is defined as the collection of paragraphs derived from a single shuffling operation; PInterleave$(\cdot)$ signifies the replacement of the original segments within the document x with the permuted paragraphs from P_{noise}; and $\tilde{x}_{\text{PInterl}}$ expresses the final output noisy document.

LInterl Noise. Based on the definitions established in Section 3.2, the process of introducing LInterl noise can be formally defined as follows:

$$LI_{\text{noise}} = \{li^1_{noise}, li^2_{noise}, \ldots, li^m_{noise}\}, \quad \tilde{x}_{\text{LInterl}} = \text{LInterleave}(x, LI_{noise}), \tag{4}$$

where LI_{noise} is defined as the collection of paragraphs derived from a single line-by-line shuffling operation. LInterleave$(\cdot)$ signifies the replacement of the original segments within the document x with the permuted paragraphs from LI_{noise}; and $\tilde{x}_{\text{LInterl}}$ expresses the final output noisy document.

To align with the empirical noise distribution observed in our statistical analysis, which reported in Appendix A, we define m as a random variable sampled from the range $\{2, 3, 4, 5\}$. This ensures that a stochastic number of noise rows are embedded into the document during each operation.

By instantiating the unified structural noise operator at different structural levels, we introduce three corresponding structural noises into the GT dataset. These noisy corpora are subsequently used as the knowledge base in the retrieval and generation stages to systematically evaluate the impact of structural noise on retrieval-augmented generation systems.

Quantification of Noise Addition To quantitatively assess the prevalence of errors within the noisy knowledge base, we calculated the CER for the knowledge base after injecting various types of noise; the average results are summarized in Table 1. The findings indicate that LInsert results in a relatively low CER. Conversely, interleaved noise leads to a sharp performance degradation, with the CER surging to over 50%. We compare these results with the error levels produced by the MinerU method to demonstrate the actual noise level we introduced. To further validate the rationality of our approach, we additionally fit the CER of documents with injected structural noise to the CER caused by real OCR and examined their overlap coefficient and distribution. The results show that the synthetic noise distribution has a high overlap coefficient with real OCR outputs(OVL = 0.7986), and the median CER values are also similar. This ensures that our benchmark neither underestimates nor exaggerates the severity of typical OCR degradation. Detailed CER results for individual documents and specific methods for fitting experiments are provided in the Appendix B.2.

Statistics	**Number (%)**
Total Documents	**425**
The New York Times (English)	244 (57.4%)
Peoples Daily (Chinese)	181 (42.6%)
Total Questions	**2,132**
NYT Subtotal	**1,186 (55.6%)**
Line Insertion Noise	378 (17.7%)
Paragraph Interleaving Noise	417 (19.6%)
Line Interleaving Noise	391 (18.3%)
PD Subtotal	**946 (44.4%)**
Line Insertion Noise	303 (14.2%)
Paragraph Interleaving Noise	301 (14.1%)
Line Interleaving Noise	342 (16.0%)

Table 2: Dataset Statistics

4.3 Q&A Pair Generation

Questions are generated using the original context in proximity to the noise injection sites to ensure

a strong coupling between the questions and the noise. To ensure high contextual dependency, we partition each context chunk used for question generation into interrelated parts A and B. Using GPT-4o (OpenAI et al., 2024), we generate questions that require joint evidence from both parts based on shared entities or semantic links. This process ultimately yields a Q&A dataset comprising three noise categories across two languages, resulting in a total of six subsets.

4.4 Quality Check

Quality is maintained through a dual-model pipeline: Qwen-Max (Yang et al., 2025) performs initial faithfulness detection, followed by a contextual dependency check. Questions answerable under degraded settings (no context, only A, or only B) are discarded, guaranteeing that the final dataset strictly requires integrated reasoning.

To ensure reliability, we evaluate answer stability by sampling each candidate ten times; questions are discarded as ambiguous unless the most frequent response appears at least four times. Finally, expert annotators conduct a manual spot check for logical soundness and discourse consistency. Human evaluation criteria is detailed in the Appendix C.2. Through this unified pipeline, we obtain a high-quality, context-dependent Q&A dataset containing 2,132 data entries. The distribution of the dataset is presented in Table 2.

5 Experiments

This section presents a comprehensive evaluation of representative LLMs and retrievers using the StruNRAG benchmark. The empirical findings address the following four research questions:

- RQ1: How does structural noise affect RAG components and model performance?
- RQ2: How does the quality check process affect the Q&A pairs?
- RQ3: Can prompt engineering effectively enhance model performance?
- RQ4: How do different retrieval parameters impact overall model performance?
- RQ5: Do structural noise and character-level noise produce non-combination effects?

5.1 Experimental Setup

To systematically assess the impact of structural noise on RAG systems, we design a set of multi-dimensional comparative experiments that isolate the effects of noise at the retrieval, generation, and end-to-end levels.

Knowledge Base. We construct multiple knowledge base variants with different quality levels for comparison. Specifically, the baselines include: (1) clean GT documents; (2) documents injected with each of the three types of synthetic structural noise.

Evaluation Metrics. We evaluate the RAG pipeline across three stages: (1) Retrieval, measured by Longest Common Subsequence (LCS) similarity; (2) Generation, using the F1-scores to assess answers; and (3) End-to-End, simulating the full pipeline and reporting F1-scores for final outputs. These F1 metrics quantify both textual overlap and answer accuracy.

Evaluated Models. To ensure the generality, we evaluate a diverse set of models covering different parameter scales and architectural choices. The evaluated LLMs, retrievers, and their end-to-end combinations are summarized in Table 3.

All open-source models are deployed in a multi-NVIDIA H800 GPU environment. A unified system prompt template is used across all experiments to eliminate the influence of prompt variations on evaluation results. In the default configuration, the retrieval depth k is set to 2.

Stage	Evaluated Models
Retrieval Stage	BGE-M3 (Chen et al., 2024) BM25
Generation Stage	Qwen2.5-72B-Instruct Llama-3.1-70B Qwen3-8B Llama-3.1-8B
End-to-End	BM25 × Qwen3-8B BM25 × Llama-3.1-8B BGE-M3 × Qwen3-8B BGE-M3 × Llama-3.1-8B

Table 3: Models evaluated at different stages of the RAG pipeline.

5.2 Impact Analysis of Structural Noise on RAG Systems (RQ1)

Table 4 demonstrates the robustness of the RAG system in dealing with structural noise.

Stage/Model		Chinese (PD)			English (NYT)		
		LInsert	PInterl	LInterl	LInsert	PInterl	LInterl
A. Retrieval (LCS)							
BGE-M3	Noisy	69.03	54.93	59.65	58.59	42.52	46.39
	GT	69.80	72.16	69.54	60.42	51.83	50.84
BM25	Noisy	**89.58**	79.63	83.27	80.36	70.72	78.78
	GT	89.53	**89.69**	**89.86**	**82.39**	**81.85**	**81.42**
B. Generation (F1)							
Qwen3-8B	Noisy	72.49	68.65	64.79	44.36	37.36	34.64
	GT	73.52	71.05	70.04	46.24	39.91	42.16
Qwen2.5-72B	Noisy	**77.16**	74.37	71.60	46.27	42.94	41.38
	GT	76.16	**75.01**	**72.87**	48.55	44.14	43.53
Llama3.1-8B	Noisy	55.32	46.85	42.55	30.47	27.57	26.28
	GT	53.89	49.85	47.73	32.84	29.51	31.10
Llama3.1-70B	Noisy	74.12	66.77	68.10	48.92	46.25	43.50
	GT	75.43	70.69	69.65	**50.56**	**47.01**	**47.76**
C. End-to-end (F1)							
BGE-M3 × Qwen3-8B	Noisy	63.42	47.30	54.74	31.42	22.19	21.38
	GT	62.91	63.90	59.28	33.65	26.64	24.16
BGE-M3 × Llama3.1-8B	Noisy	53.22	38.88	39.16	24.55	20.09	17.54
	GT	52.83	51.27	45.58	27.12	23.17	20.28
BM25 × Qwen3-8B	Noisy	73.26	64.18	63.90	41.01	36.29	31.23
	GT	**74.57**	**73.15**	**70.67**	**42.85**	**37.95**	**36.91**
BM25 × Llama3.1-8B	Noisy	61.54	50.27	45.24	31.44	28.16	27.37
	GT	62.43	57.12	57.08	35.11	32.33	28.12

Table 4: The impact of three categories of structural noise on RAG systems. Performance at the retrieval, generation, and end-to-end stages is reported for both noisy and GT settings. The best results of each stage are highlighted in bold.

Retrieval results. Overall, retrieval models perform poorly in the presence of structural noise, suffering over 10% performance loss. BGE-M3 is particularly sensitive: in the Chinese PInterl task, its score plummeted from 72.16 to 54.93. In contrast, BM25 remained more stable. We conducted experiments to explore the reasons for the superior performance of sparse retrieval, and detailed results are reported in Appendix E. Additionally, lower scores in PInterl and LInterl compared to LInsert suggest that cross-line or cross-paragraph interleaving challenges retrieval more than simple noise insertion.

Generation results. During the generation stage, the LLMs remain relatively stable, with larger scales (70B/72B) exhibiting superior robustness. Under LInsert, certain models even match or exceed GT performance, suggesting an inherent noise-filtering capacity. Chinese tasks consistently yield higher F1-scores than English ones, likely reflecting more concentrated themes and easier reasoning within the Chinese corpus. Furthermore, Qwen and Llama excel in Chinese and English subsets respectively, underscoring distinct cross-lingual performance disparities.

End-to-end results. Retrievers define the performance ceiling of RAG systems. We observe that LLMs × BGE-M3 consistently underperform relative to the standalone generation stage, suggesting that LLMs cannot compensate for retrieval failures caused by retriever fragility. Conversely, the LLMs × BM25 configuration demonstrates superior robustness under structural noise, with end-to-end performance occasionally surpassing the generation stage. This indicates that an effective retriever, by isolating relevant context, can yield better results than providing the entire context.

More RAG strategies. We further evaluated hybrid retrieval and re-ranking strategies based on RRF, and employed the BGE-Reranker-v2-M3 model[3] as a cross-encoder re-ranker. Experiments were conducted using Qwen3-8B as the generation

[3]https://huggingface.co/BAAI/bge-reranker-v2-m3

Strategy		English (NYT)		
		LInsert	PInterl	LInterl
BGE-M3	GT	33.65	26.64	24.16
	Noise	31.42	22.19	21.38
BM25	GT	**42.85**	**37.95**	**36.91**
	Noise	41.01	36.29	31.23
BM25 + BGE-M3	GT	33.71	29.47	27.87
	Noise	31.92	25.94	25.12
BGE-M3 × Reranker	GT	32.74	25.56	23.07
	Noise	28.84	22.06	21.38

Table 5: Hybrid retrieval and re-ranking performance.

Model		English (NYT)		
		LInsert	PInterl	LInterl
Vision-Language Models				
Qwen2.5-VL-7B		42.15	38.93	38.91
Qwen2.5-VL-32B		34.85	38.52	41.08
RAG Systems				
Qwen3-8B	Noisy	44.36	37.36	34.64
	GT	46.24	39.91	42.16
Qwen3-72B	Noisy	46.27	42.94	41.38
	GT	48.55	44.14	43.53

Table 6: Comparison between VLMs and RAG systems.

model, and the results are shown in Table 5. After introducing the re-ranker, the model performance decreases under both the GT and noise settings. Hybrid retrieval (BM25 + BGE-M3) outperforms purely dense retrieval, but in certain noisy scenarios it still falls short of pure BM25.

VLM performance. We further conducted experiments using VLMs to compare their performance with RAG systems under structural noise. Specifically, we evaluated the latest Qwen2.5-VL series models (7B and 32B).

The results are shown in Table 6. For smaller models, VLMs perform worse than RAG under the GT setting. However, under noisy conditions, Qwen2.5-VL-7B outperforms Qwen3-8B on PInterl and LInterl noise types. For larger models, VLMs perform worse than RAG in both GT and Noise settings.

5.3 Study of Data Quality Check Process (RQ2)

We conducted ablation studies to validate the rationality of the Q&A's quality check process to ensure the high quality of our dataset construction pipeline, with the results presented in Table 7. Specifically, -fai denotes the removal of the faithfulness detection step, while -rel indicates the exclusion of the contextual dependency mechanism. We observe that removing faithfulness detection increases difficulty but compromises quality by introducing irregular question formats. Conversely, eliminating the contextual dependency check simplifies the questions. Detailed case study is provided in Appendix B.1.

Generation/Model		English (NYT)		
		LInsert	PInterl	LInterl
Qwen3-8B	-fai	36.28	44.70	29.39
	-rel	60.18	64.03	57.12
	$full$	44.35	37.36	34.26

Table 7: Ablation Study of StruNRAG Dataset Construction.

5.4 Study of Prompting Strategies (RQ3)

While prompt engineering is typically effective in guiding models to mitigate noise, our findings indicate that this does not consistently hold true in the presence of structural noise. The results of the prompt ablation study are presented in Table 8. Specifically, Qwen2.5-72B and Llama3.1-8B demonstrate substantial improvements, suggesting that prompting effectively guides these models to distinguish or disregard noise. Conversely, both Llama3.1-70B and Qwen3-8B exhibit performance degradation when prompting is applied compared to the raw noisy baseline. We hypothesize that the additional prompts may introduce excessive constraints, thereby disrupting the models' inherent reasoning chain. The prompt setting is in Appendix F.

Generation/Model		English(NYT)		
		LInsert	PInterl	LInterl
Qwen3-8B	Noisy	**44.36**	**37.36**	**34.64**
	w/Prompt	42.58	36.10	33.82
Qwen2.5-72B	Noisy	46.27	42.94	41.38
	w/Prompt	**47.99**	**45.25**	**43.67**
Llama3.1-8B	Noisy	30.47	27.57	26.28
	w/Prompt	**34.21**	**27.75**	**29.07**
Llama3.1-70B	Noisy	**48.92**	**46.25**	**43.50**
	w/Prompt	46.01	43.81	42.43

Table 8: Ablation Study of Prompting Strategies for Handling Noisy Contexts.

5.5 Hyper-parameter Settings (RQ4)

We evaluate the impact of different retrieval depths ($k \in \{1, 2, 5\}$) on the system under both Noisy and GT settings. The experimental results are shown in Figure 4. Increasing the number of retrieved documents (k) exhibits a trend of diminishing returns. While increasing k from 1 to 2 generally improves performance, extending to Top-5 often leads to performance stagnation or even regression. This implies that while a larger k value improves recall, it also introduces excessive noise that overwhelms the generator, verifying the trade-off between context recall and noise tolerance. Detailed results are provided in Appendix D.

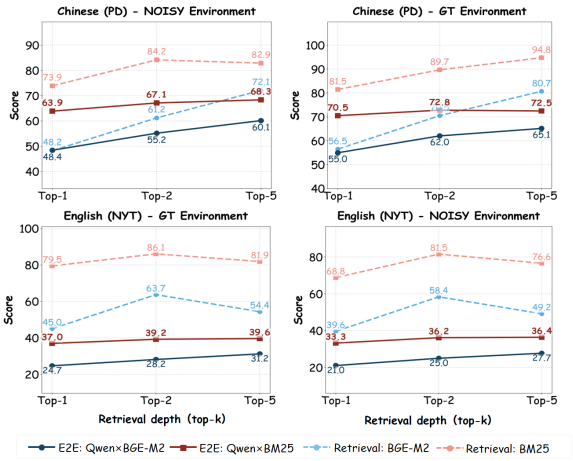

Figure 4: Impact of retrieval depth (k) on system performance. The vertical axis indicates the average score across three interference settings.

5.6 Non-compositional Effects of Structural and Character-level Noise(RQ5)

Real-world OCR outputs typically contain both structural distortions and character-level recognition errors. In this work, we isolate structural noise and quantify its independent impact on the RAG pipeline. When structural noise coexists with character corruption, retrieval matching may be further disrupted and the generation model's ability to recover from disordered structure may be weakened. To investigate this phenomenon, we construct noisy knowledge bases by combining three types of structural noise with character-level noise and evaluate them using our dataset on the English NYT benchmark. The results are shown in Table 9.

The results demonstrate clear non-compositional effects between structural and character-level noise, and the direction of deviation is inconsistent. Specifically, character noise combined with LInsert noise produces a synergistic degradation, amplifying performance loss. In contrast, character noise combined with PInterl and LInterl exhibits an inhibitory interaction, where structural noise dominates and the marginal impact of character noise is reduced.

Stage / Setting	English (NYT)		
	LInsert	PInterl	LInterl
Retrieval (BGE-M3)			
GT	60.42	51.83	50.84
Char	56.95	50.92	48.03
StruNoise	58.59	42.52	46.39
Char+StruNoise	53.04	42.16	47.07
Generation (Qwen3-8B)			
GT	46.24	39.91	42.16
Char	45.18	36.06	37.02
StruNoise	44.36	37.36	34.64
Char+StruNoise	40.83	36.03	32.19
End-to-end (BGE-M3 × Qwen3-8B)			
GT	33.65	26.64	24.16
Char	31.13	24.95	22.90
StruNoise	31.42	22.19	21.38
Char+StruNoise	28.50	21.46	20.44

Table 9: Non-compositional effects between character-level and structural noise.

6 Conclusion

This paper identifies and formalizes a critical yet often overlooked bottleneck in real-world Retrieval-Augmented Generation (RAG) systems: OCR-induced structural noise. To systematically evaluate this challenge, we introduce the StruN-RAG benchmark.

Our comprehensive evaluation yields several key findings. First, structural noise triggers cascading performance degradation across the RAG pipeline, and commonly adopted RAG optimization strategies do not consistently improve robustness. Second, visionlanguage models (VLMs) exhibit limited effectiveness when handling documents with complex layouts. We further observe that simple prompt engineering is not a universal remedy for structural noise, as its effectiveness is highly model-dependent. In addition, increasing the retrieval depth (k) provides diminishing returns and may even introduce excessive noise. Finally, structural noise and character-level noise exhibit non-compositional effects, highlighting that evaluating their interaction remains an important direction for future research.

Limitations

Despite the comprehensive design of StruNRAG, we acknowledge several limitations in this study. First, while our noise injection methodology is grounded in statistical analysis of real OCR errors, the structural noise in our dataset is synthetically generated. Although this allows for controlled variable testing, it may not perfectly capture the erratic distribution of errors found in diverse real-world parsing engines. Second, our document corpus is primarily composed of news media (The New York Times and People's Daily). While these sources offer complex layouts, they may not fully represent domains like scientific literature or financial reports, which present unique challenges through charts and formulas. Finally, the non-compositional effects of character-level noise and structural noise on RAG systems are an interesting area for investigation, but this paper was unable to conduct large-scale and more fine-grained experiments in this area.

Ethics Statement

Data Copyright and Licensing The datasets used in this research are derived from public news sources. We conducted a strict manual review process to identify and neutralize any sensitive, offensive, or harmful content. This ensures that all published materials adhere to ethical standards and content regulations.

Human Annotation Our annotation process involved four human annotators who reviewed the model-generated outputs. To ensure fair compensation, annotators were paid an average of $12 per hour, which aligns with universal ethical standards for crowdsourcing and contract work. All annotators participated voluntarily and were fully informed of the nature and objectives of the task.

References

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural optical understanding for academic documents. *Preprint*, arXiv:2308.13418.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, Xihui Liu, and Hongsheng Li. 2025. Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *Preprint*, arXiv:2503.10639.

Rujun Han, Yuhao Zhang, Peng Qi, Yumo Xu, Jenyuan Wang, Lan Liu, William Yang Wang, Bonan Min, and Vittorio Castelli. 2024. RAG-QA arena: Evaluating domain robustness for long-form retrieval augmented question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4354–4374, Miami, Florida, USA. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Chen Li. 2024. gptpdf: Using GPT-4o to parse PDF to Markdown. https://github.com/CosmosShadow/gptpdf. Accessed: 2025-05-20.

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.

Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. 2023. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):919–931.

Xin Lv, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Yichi Zhang, and Zelin Dai. 2021. Is multi-hop reasoning really explainable? towards benchmarking reasoning interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8899–8911, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. 2025. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24838–24848.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 37433751, New York, NY, USA. Association for Computing Machinery.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. ARES: An automated evaluation framework for retrieval-augmented generation systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 338–354, Mexico City, Mexico. Association for Computational Linguistics.

Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. 2021. Layoutparser: A unified toolkit for deep learning based document image analysis. In *Document Analysis and Recognition ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 510, 2021, Proceedings, Part I*, page 131146, Berlin, Heidelberg. Springer-Verlag.

Baoguang Shi, Xiang Bai, and Cong Yao. 2017. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304.

Vikrant Varma and 1 others. 2023. Marker: High-speed, high-quality PDF to Markdown conversion. https://github.com/datalab-to/marker. Accessed: 2025-06-01.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. 2024a. Mineru: An open-source solution for precise document content extraction. *Preprint*, arXiv:2409.18839.

Jiawei Wang, Kai Hu, and Qiang Huo. 2024b. Dlaformer: An end-to-end transformer for document layout analysis. In *Document Analysis and Recognition - ICDAR 2024: 18th International Conference, Athens, Greece, August 30September 4, 2024, Proceedings, Part IV*, page 4057, Berlin, Heidelberg. Springer-Verlag.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online. Association for Computational Linguistics.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 11921200, New York, NY, USA. Association for Computing Machinery.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao

Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. mplug-docowl: Modularized multimodal large language model for document understanding. *Preprint*, arXiv:2307.02499.

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.

Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. 2024. Llavar: Enhanced visual instruction tuning for text-rich image understanding. *Preprint*, arXiv:2306.17107.

Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. EAST: An Efficient and Accurate Scene Text Detector . In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2642–2651, Los Alamitos, CA, USA. IEEE Computer Society.

Anni Zou, Wenhao Yu, Hongming Zhang, Kaixin Ma, Deng Cai, Zhuosheng Zhang, Hai Zhao, and Dong Yu. 2025. DocBench: A benchmark for evaluating LLM-based document reading systems. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pages 359–373, Albuquerque, New Mexico, USA. Association for Computational Linguistics.

Appendix

A Motivation Experiments

Noise statistics are illustrated in Figure A.1, where (a) represents the proportion of CER character noise and (b) denotes the amount of structural noise. Statistical analysis reveals that the frequency of noise occurrences is primarily concentrated between 2 and 5, accounting for over 90% of the documents in the dataset. Regarding the Character Error Rate (CER), the distribution exhibits a peak within the [60%, 80%] interval, while other intervals remain relatively sparse.

B Details of StruNRAG Benchmark Construction

B.1 Case Study

Figure B.1 illustrates the presence of three types of noise in the actual dataset and how noise actually affects each stage of RAG. In this example,

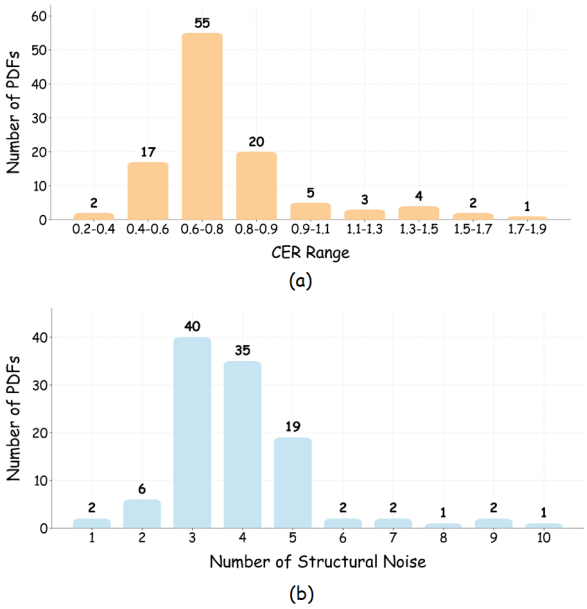

Figure A.1: Noise statistics

while LInsert noise led to the insertion of irrelevant content, it did not affect the correctness of the retrieval and answer generation stages. PInterl noise resulted in incomplete retrieved context, making the question unanswerable. In the LInterl noise scenario, although relevant text was retrieved, the interleaved content caused irrelevant names to be appended to the question, leading the model to generate an incorrect answer.

Figure B.2 illustrates how unacceptable questions were eliminated during the quality check phase. The standards for manually selecting question quality will be further explained in the appendix.

B.2 Empirical Analysis of Structural Noise in Complex Layouts

To quantitatively assess the extent of impact from structural noise, we isolated specific noise categories within the PD and NYT datasets, retaining different structural perturbations independently. The results indicate that while LInsert results in a relatively low CER of approximately 11-12%, interleaving noiseparticularly PInter and LInterltriggers a drastic performance drop, with CER soaring above 50%. The CER distributions for each dataset are illustrated in Figure B.3.

In real-world OCR scenarios, the proportions of the three types of structural noise are not identical. In our main experiments, we set their intensities to the same range in order to quantitatively compare their individual impacts. We now further combine


```
{
  "doc_name": "news/12_28",
  "ID": "c563a1f4-533d-4305-8bf0-",
  "questions": "Who released the body camera footage of corrections officers attacking Robert Brooks in a New York prison?",
  "answers": "Letitia James",
  "doc_type": "news",
  "questions_type": "LInsert",
  "evidence_page_no": 0,
  "evidence_source": "Corrections officers punch... It was made public by Letitia James"
},
```


Corrections officers punched and kicked a handcuffed, shackled inmate in the groin and chest during a fatal attack at a prison in central New York this month, video footage released publicly on Friday shows.

The footage was recorded by body cameras worn by four of the officers. It was made public by Letitia James, the state attorney general, as part of her office's investigation into the death of the man, Robert Brooks, and the beating that preceded it.

LInsert

Noise Base: Corrections officers punched ... on Friday shows. The Inquiries Into Death at New York Facility ... Robert Brooks, and the beating that preceded it.

Retrieval: Corrections officers punched ... on Friday shows. The Inquiries Into Death at New York Facility ... Robert Brooks, and the beating that preceded it.

Answer: **Letitia James**

PInterl

Noise Base: The footage was recorded by body cameras... preceded it. Corrections officers punched and kicked a handcuffed, shackled ... on Friday shows.

Retrieval: Corrections officers punched and kicked a handcuffed, shackled inmate in the groin and chest during a fatal attack at a prison in central New York this month, video footage released publicly on Friday shows.

Answer: **Not answerable.**

LInterl

Noise Base: ... in the groin and chest during a fatal attack at a prison in cen- vestigation into the death of the tral New York this month, video man, Robert Brooks, and the beat footage released publicly on Fri ing that preceded it. day shows.

Retrieval: ... in the groin and chest during a fatal attack at a prison in cen- vestigation into the death of the tral New York this month, video man, Robert Brooks, and the beat footage released publicly on Fri ing that preceded it. day shows.

Answer: **Robert Brooks.**

Figure B.1: A case using noisy knowledge base as the evidence source on a news document.

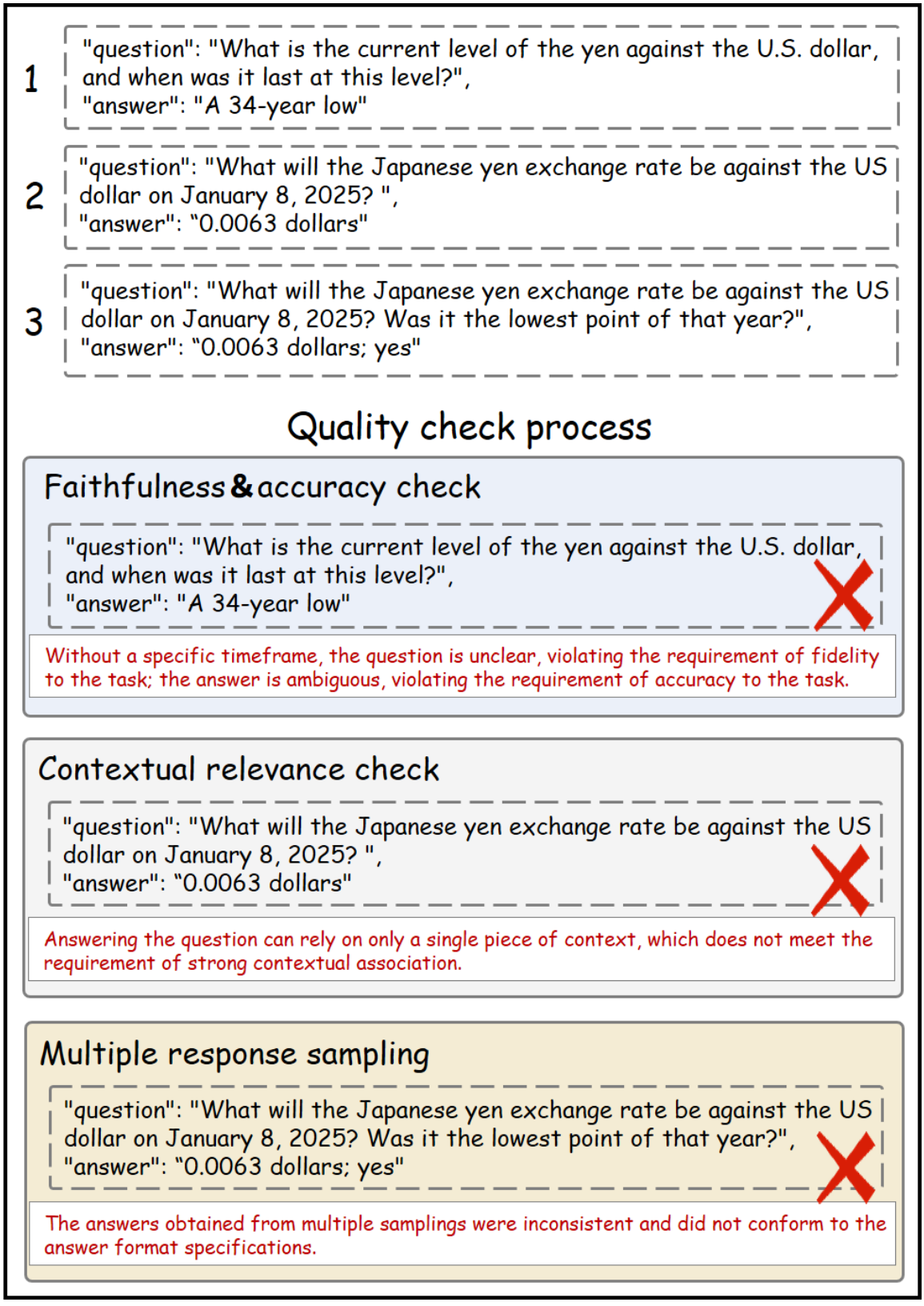

Figure B.2: A case for Q&A quality check.

the three noise types according to their real-world proportions to simulate a mixed-noise setting.

We model the mixed CER using the following formulation:

$$\mathrm{C}_{\mathrm{Mix}} = \mathrm{C}_{\mathrm{PInterl}} + \beta \cdot \mathrm{C}_{\mathrm{LInterl}} + \alpha \cdot \mathrm{C}_{\mathrm{LInsert}} + \epsilon$$

where the parameters $\beta = 1.1$ and $\alpha = 0.4$ are determined based on the observed error frequencies.

The results show that the synthetic noise distribution has a high overlap coefficient with real OCR outputs, with $\mathrm{OVL} = 0.7986$, as illustrated in Figure B.4. The median CER values are also highly consistent (0.72 vs. 0.69). Such consistency across both Chinese and English corpora suggests that our simulation of structural noise aligns well with the actual noise levels introduced during practical OCR processes.

We further conduct a comparative experiment in NYT dataset to examine whether the RAG system exhibits similar performance degradation under different noise conditions. The results are shown in Table B.1. When each noise type is applied individually, the performance degradation of the RAG system is smaller than that observed under real OCR conditions. However, when the three noise types are superimposed, a non-compositional effect emerges, leading to amplified performance degradation rather than a simple additive drop. Moreover, this degradation trend closely matches the behavior observed with real OCR outputs. These results further confirm that our noise simulation realistically reflects practical OCR scenarios.

B.3 Q&A Generation Details

To ensure high-quality QA generation, we designed specialized prompts for both generation and verification (see Table B.2 and Table B.3). For Chinese instances, language-specific templates and instructions were utilized to guarantee linguistic consistency in model outputs. Our rigorous filtering pipeline began with an initial pool of 6,840 questions. Following a faithfulness check, 5,688 questions were retained. This set was further pruned to 2,332 after applying contextual dependency filters. Finally, through iterative sampling and expert auditing, we curated a dataset of 2,132 high-quality QA pairs.

C Human Evaluation Criteria

The manual filtering and data annotation were conducted by native speakers with higher education backgrounds residing in China. All annotators underwent specialized training to standardize the labeling workflow, and possess expertise in NLP, ensuring that the identification of sensitive content aligns with local linguistic and cultural norms.

C.1 Data Collection Evaluation

To ensure the high fidelity of the dataset, we established a rigorous set of labeling guidelines for the manual verification phase. Annotators were required to: (1) rectify all character-level OCR errors, including glyph confusion and punctuation mismatches; (2) strictly enforce the correct linearization of text flows across columns and pages, resolving any misordering issues arising from automated parsing; (3) reconstruct broken sentences by removing erroneous line breaks and addressing end-of-line hyphenation; and (4) properly handle non-narrative elements, such as merging stylized drop caps back into their corresponding words. This standardized workflow ensures that the resulting text framework faithfully reflects the inherent information flow of the original documents while minimizing structural bias.

C.2 Quality Check Evaluation

To ensure the reliability and logical validity of our QA dataset, we implemented a rigorous expert review process for manual inspection. Expert annotators were tasked with auditing a randomly sampled subset of the LLM-generated QA pairs based on four key dimensions: (1) ensuring each answer is strictly grounded in the source text, free from any extrinsic hallucinations; (2) verifying that questions necessitate document-specific reading and cannot be answered solely through common sense or parametric knowledge; (3) confirming that questions require information synthesis rather than simple keyword matching; and (4) checking for grammatical fluency and the absence of ambiguous pronouns. Finally, a random spot check was performed on the final results to further validate data quality.

D Impact of Retrieval Depth (Top-k) on Structural Noise Robustness

Table D.1 presents the performance dynamics of both the retrieval module and the end-to-end RAG pipeline across varying retrieval depths ($k \in \{1, 2, 5\}$), specifically under distinct OCR-induced structural perturbations (LInsert, PInterl,

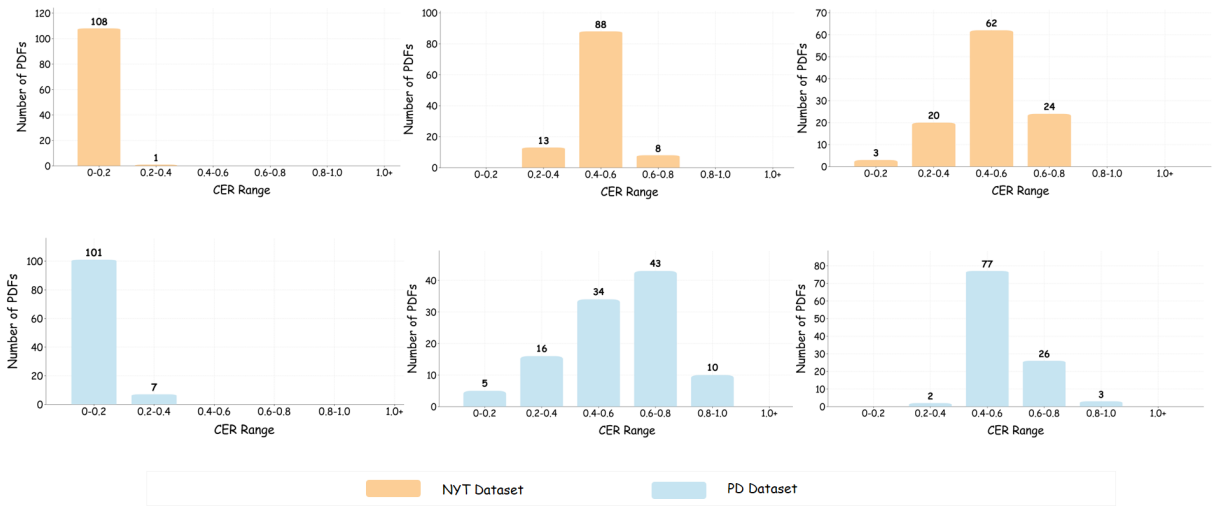

Figure B.3: CER of noise base

Model	GT	OCR	Composite	Superimposed	LInsert	PInterl	LInterl
BGE-M3	64.87	47.35	50.36	42.84	62.97	60.80	62.32
Qwen3-8B	69.60	54.35	56.54	52.54	69.71	67.83	67.07
BGE-M3 × Qwen3-8B	57.89	46.16	48.50	42.41	56.91	52.92	55.90

Table B.1: RAG performance under real OCR and simulated noise settings

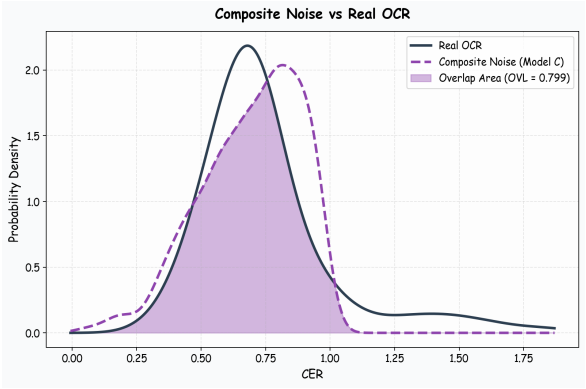

Figure B.4: Fit of injected noise to actual OCR results (CER)

and LInterl). Our analysis unveils a nuanced interplay among retrieval granularity, noise topology, and system robustness.

D.1 Retrieval Stage Analysis

In the retrieval phase, evaluated using the LCS metric, increasing the retrieval depth k universally improves the tolerance of the system to structural noise, although it does not fully bridge the performance gap induced by such perturbations. As anticipated, expanding k from 1 to 5 yields a monotonic improvement in retrieval performance across both GT and Noisy settings. Specifically, on the PD dataset, BM25's performance under LInsert noise climbs from 81.16 at Top-1 to 86.74 at Top-5. This suggests that widening the candidate document window enables the retriever to capture relevant segments that were potentially down-weighted due to structural dislocation, thereby partially compensating for the feature mismatch caused by OCR noise. Notably, sparse retrievers appear to derive greater benefit from increased Top-k depths compared to dense retrievers when mitigating structural noise. For instance, under the Top-1 NYT PInterl setting, BGE-M3 achieves a score of only 34.68, significantly lagging behind the GT baseline (45.04); this disparity remains substantial even at Top-5 (42.52 vs. 51.83). This corroborates our hypothesis that structural noise consistently degrades retrieval performance, and merely increasing k cannot fundamentally resolve the semantic vector shifts or keyword context disruption triggered by severe layout distortions.

D.2 End-to-end Analysis

In our end-to-end experiments, we evaluate the generator's performance across varying amounts of retrieved context using the F1-score. The results demonstrate that the generation stage ex-

System:

You are a helpful assistant that generates high-quality question-answer pairs from documents for RAG tasks. Always return valid JSON format as requested.

User:

You are an AI language model specialized in generating structured question-answer (QA) pairs from document texts for use in retrieval-augmented generation (RAG) tasks.
Your task is to generate RAG-style question-answer pairs from a given document chunk. Each chunk is divided into two parts, **A** and **B**, separated by `SEGMENTATION POSITION`.
To generate such question-answer pairs, you first need to read the first and last parts of the given chunk (A, B), identify text containing the same entities, and formulate a question-answer pair based on the content of A and B. It is impossible to answer the question based solely on information from A or B.

Requirements:
1. The question should be self-contained and understandable without access to the document.
2. Avoid any references to page numbers, figures, or "in the document" phrasing.
3. Questions should not quote document text verbatim; instead, rewrite naturally.
4. Avoid ambiguous pronouns like "he", "she", or "it".
5. Generate exactly one question-answer pair.
6. CRITICAL: The answer must be concise and specific with a fixed, accurate answer. Answer length should not exceed 10 tokens.
7. Avoid overly open-ended questions and lengthy descriptive answers.
8. Focus on factual questions with precise answers like names, dates, locations, numbers, or specific terms.
Each QA must include: `question`, `answer` (max 10 tokens), and `evidence_context`.
Return only a JSON object (not an array) with the QA pair. Do not include explanations or preambles.

{qa_examples}
`<document>`{document}`</document>`

Table B.2: Prompt for NYT dataset QA generation.

hibits non-trivial robustness, and the choice of Top-k is critical for mitigating cascading errors. By comparing Top-1 and Top-5 results, we observe that LLMs possess a strong capability to extract accurate information from noisy contexts. As k increases, the performance gains in the end-to-end setting are generally more pronounced than those in the retrieval stage alone. For instance, under the English (NYT) LInsert noise setting, the performance of Qwen3-8B $\times$ BM25 improves from 37.93 (Top-1) to 41.83 (Top-5), approaching the performance achieved with GT. However, although increasing k generally boosts F1-scores, the benefits show a diminishing trend when facing severe reading order disruptions. When input documents contain severe line-level interleaving, merely aggregating more context (Top-5) may introduce excessive irrelevant noise segments, thereby saturating the LLM's in-context learning capabilitie.

E In-depth Analysis of Retrieval Models Performance

To explore the reasons for the performance differences among different retrieval models, we conducted an embedding drift analysis under different structural noise intensities. The experimental results are shown in Table E.1.

Jaccard similarity is a set-based discrete metric. The experimental results show a constant

System:

You are an AI specialized in document question-answering verification. Your mission is to analyze the given question-answering pairs and follow the instructions. Your response must be true and accurate, and no additional content should be output.

1. Question Type Check

Does the question match the task description: {DETAILED_TASK_DESCRIPTION} Make sure the question meets the required task context.

2. Evidence Relevance Check

Does the provided evidence context relate to the question provided? Does the answer accurately reflect the information in the evidence context? Ensure the question is formulated based on information explicitly stated. The question should not introduce concepts unrelated to the document's content.

3. Clarity and Precision

Is the question clear and unambiguous? And is the answer concise and precise? Ensure the language is straightforward and easily understandable, and avoid complex phrasing that may confuse the reader. The intention of the question and answer pair must be clear and direct, avoiding verbosity and unnecessary detail. Ensure the answer fully addresses the question without omitting crucial information.

Please provide ONLY your verification result in JSON format. Do not include explanations, reasons, scores, or recommendations. Use exactly this structure:

```
{
  "question_type_check":
  "evidence_relevance_check":
  "clarity_precision_check":
}
```

Return only a JSON object (not an array) with the QA pair. Do not include explanations or preambles.

`Input:` {question & answer}

Table B.3: Prompt for NYT dataset QA check.

value of 1.0, indicating that structural noise does not change the lexical composition of the text; therefore, the performance of the BM25 retrieval model is minimally affected. BGE-M3 is based on a Transformer architecture. Even when lexical units remain unchanged, structural noise interferes with positional encoding, causing semantic shifts in the representation space. As the noise intensity increases, the embedding similarity of BGE-M3 continuously decreases, indicating that structural noise mainly disrupts semantic representation rather than term matching. Consequently, dense retrieval models exhibit more significant performance degradation compared to sparse retrieval models.

F Q&A Instruction Prompt Setting

The template is shown in Table F.1. For the same task, we strictly use the same prompt to avoid the impact of prompt changes on the results.

			Chinese(PD)			English(NYT)		
			LInsert	PInterl	LInterl	LInsert	PInterl	LInterl
Top-1								
Retrieval (LCS)	BGE-M3	Noise	55.79	43.14	45.81	46.56	34.68	37.70
		GT	56.93	57.88	54.67	47.66	45.04	42.38
	BM25	Noisy	81.16	66.94	73.50	71.87	62.90	71.67
		GT	82.89	80.91	80.62	92.64	71.44	74.28
End-to-end (F1)	Qwen3-8B × BGE-M3	Noisy	56.46	42.03	46.78	25.74	18.21	19.19
		GT	57.62	55.07	52.21	29.09	23.24	21.70
	Qwen3-8B × BM25	Noisy	70.64	60.45	60.53	37.93	30.10	31.75
		GT	72.43	69.54	69.53	40.78	34.71	35.46
Top-2								
Retrieval (LCS)	BGE-M3	Noise	69.03	54.93	59.65	67.09	52.20	55.79
		GT	69.80	72.16	69.54	69.84	62.17	59.14
	BM25	Noisy	89.58	79.63	83.27	84.49	76.60	83.56
		GT	89.53	89.69	89.86	86.51	85.19	86.50
End-to-end (F1)	Qwen3-8B × BGE-M3	Noisy	63.42	47.30	54.74	31.42	22.19	21.38
		GT	62.91	63.90	59.28	33.65	26.64	24.16
	Qwen3-8B × BM25	Noisy	73.26	64.18	63.90	41.01	36.29	31.23
		GT	74.57	73.15	70.67	42.85	37.95	36.91
Top-5								
Retrieval (LCS)	BGE-M3	Noise	83.28	73.30	59.65	58.59	42.52	46.39
		GT	78.90	81.40	81.81	60.42	51.83	50.84
	BM25	Noisy	86.74	73.30	88.56	80.36	70.72	78.78
		GT	94.16	95.23	94.86	82.39	81.85	81.42
End-to-end (F1)	Qwen3-8B × BGE-M3	Noisy	67.80	55.20	57.39	33.34	26.95	22.87
		GT	65.49	66.31	63.63	37.29	29.77	26.58
	Qwen3-8B × BM25	Noisy	73.73	67.52	63.77	41.83	34.82	32.54
		GT	74.96	71.99	70.44	42.61	38.52	37.74

Table D.1: Result of Different k Values.

Noise Level (n)	**BGE Cosine Similarity**	**BGE Similarity Drop**	**Jaccard Similarity**
2	0.9178	0.0822	1.0000
3	0.8949	0.1051	1.0000
4	0.8904	0.1096	1.0000
5	0.8786	0.1214	1.0000

Table E.1: Impact of Structural Noise on Retrieval Similarity

Raw Prompt	You are an expert, you have been provided with a question and documents retrieved based on that question. Your task is to search the content and answer these questions using both the retrieved information. You **MUST** answer the questions briefly with one or two words or very short sentences, devoid of additional elaborations. Write the answers within `<response></response>`. If you cannot find answer from retrieved Documents, say: "Not answerable".
Correction Prompt	You are an expert, you have been provided with a question and documents retrieved based on that question. Your task is to search the content and answer these questions using both the retrieved information. There are some order errors in these documents. Please ignore them for now, try to restore the document logic, and then answer. You **MUST** answer the questions briefly with one or two words or very short sentences, devoid of additional elaborations. Write the answers within `<response></response>`. If you cannot find answer from retrieved Documents, say: "Not answerable".

Table F.1: Q&A Instruction Prompt