

LongRAG: A Dual-Perspective Retrieval-Augmented Generation Paradigm for Long-Context Question Answering

Qingfei Zhao^{1,2,†}, Ruobing Wang^{1,2}, Yukuo Cen⁴,
Daren Zha¹, Shicheng Tan³, Yuxiao Dong³, Jie Tang^{3,*}

¹Institute of Information Engineering, Chinese Academy of Sciences;

²School of Cyber Security, University of Chinese Academy of Sciences;

³Tsinghua University; ⁴Zhipu AI

{zhaoqingfei, wangruobing, zhadaren}@iie.ac.cn, yukuo.cen@zhipuai.cn
tsctan@foxmail.com, {yuxiaod, jietang}@tsinghua.edu.cn

Abstract

Long-Context Question Answering (LCQA), a challenging task, aims to reason over long-context documents to yield accurate answers to questions. Existing long-context Large Language Models (LLMs) for LCQA often struggle with the "lost in the middle" issue. Retrieval-Augmented Generation (RAG) mitigates this issue by providing external factual evidence. However, its chunking strategy disrupts the global long-context information, and its low-quality retrieval in long contexts hinders LLMs from identifying effective factual details due to substantial noise. To this end, we propose LongRAG, a general, dual-perspective, and robust LLM-based RAG system paradigm for LCQA to enhance RAG's understanding of complex long-context knowledge (i.e., global information and factual details). We design LongRAG as a plug-and-play paradigm, facilitating adaptation to various domains and LLMs. Extensive experiments on three multi-hop datasets demonstrate that LongRAG significantly outperforms long-context LLMs (up by 6.94%), advanced RAG (up by 6.16%), and Vanilla RAG (up by 17.25%). Furthermore, we conduct quantitative ablation studies and multi-dimensional analyses, highlighting the effectiveness of the system's components and fine-tuning strategies. Data and code are available at <https://github.com/QingFei1/LongRAG>.

1 Introduction

Large language models (LLMs), such as GPT (Brown et al., 2020), GLM (Zeng et al., 2022) and LLaMA (Touvron et al., 2023), boost the real-world development of multiple scenarios. Long-context question answering (LCQA) (Caciularu et al., 2022), which has been recently advanced significantly by LLMs, is a complex task that requires reasoning over a long document or multiple documents to provide accurate answers to questions. Re-

*Corresponding author

†Work done when QZ interned at Zhipu AI

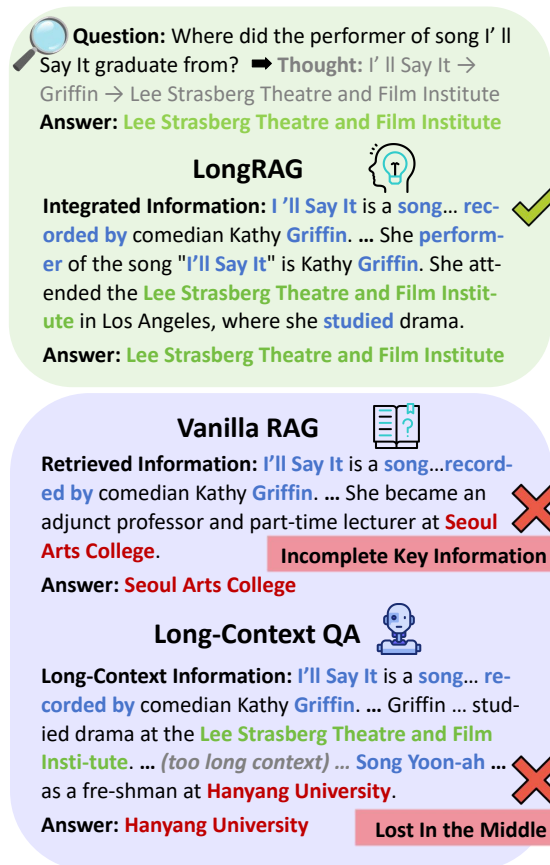


Figure 1: **Examples of Different Methods.** Long-Context LLMs and Vanilla RAG face "lost in the middle" and "incomplete key information" issues, while LongRAG addresses them, yielding a perfect answer.

cently, several long-context LLMs have been introduced, such as Gemini (Anil et al., 2023) and GPT-4-128k, capable of ingesting entire relevant documents and generating answers directly. However, as shown in Figure 1, they frequently encounter the "lost in the middle" issue (Liu et al., 2024), that is, when the relevant context is in the middle of the document (rather than the beginning and end), they are prone to sub-optimal or even incorrect responses. Instead, the Retrieval-Augmented Generation (RAG) system (Gao et al., 2023; Guu et al.,

2020) offers an alternative approach, mitigating this issue by employing a fixed-length chunking strategy (Theja, 2023). This strategy ensures the input to the LLM is concise and highly relevant to the question.

Nevertheless, Vanilla RAG remains insufficient for the LCQA task due to two major limitations. **First**, the chunking strategy disrupts the contextual structure and background information in long documents (global information). Some chunks may contain incomplete information (Dong et al., 2023), thereby causing LLMs to draw upon irrelevant context or fall back on their internal parameterized knowledge, potentially leading to inaccurate responses. **As depicted in Figure 1**, Vanilla RAG only retrieves "Griffin" as the performer of "I'll say it" but misses the university from which "Griffin" graduated. Although the "university" is mentioned in the same paragraph, the system ultimately produces an incorrect response. **Second**, low evidence density in long-context documents can lead to low retrieval quality. Considerable noise present in long-context documents impairs LLMs' capacity to accurately identify key information (factual details), resulting in the retrieval of low-quality chunks and ultimately leading to erroneous answers (Zhang et al., 2023; Chen et al., 2024). Recently, several advanced RAG systems have attempted to mitigate the aforementioned issues. Specifically, Self-RAG (Asai et al., 2023) employs self-reflection tokens to facilitate the autonomous exploration of global information in a corpus. However, its reliance on the accuracy of reflection tokens may result in the potential deletion of valid retrieval chunks with factual details. CRAG (Yan et al., 2024) evaluates the question relevance of each chunk individually to enhance the identification of factual details. Nevertheless, it overlooks the connections between chunks, provoking low-quality evaluation when valid details span multiple chunks, potentially leading to the omission of crucial factual details.

In our work, we propose LongRAG, a general, dual-perspective, and robust RAG system paradigm that effectively addresses the above-mentioned issues for LCQA, comprising four plug-and-play components with multiple strategies: a hybrid retriever, an LLM-augmented information extractor, a CoT-guided filter, and an LLM-augmented generator. LongRAG enhances the RAG system's ability to mine global long-context information and identify factual details. Specifically, the long-

context extractor employs a mapping strategy to orderly extend the semantic space of retrieved chunks into a higher dimensional long-context semantic space, then refining global information and contextual structure among chunks. Meanwhile, the CoT-guided filter utilizes the Chain of Thought (CoT) (Wei et al., 2022) to provide global clues according to the knowledge of all retrieved chunks, instructing LLMs to carefully review factual details and precisely filter out irrelevant chunks. This improves evidence density and enhances RAG's ability to understand complex and lengthy contexts. Additionally, we have curated an automated instruction data pipeline for constructing a high-quality dataset for fine-tuning. This fine-tuning strategy significantly enhances the "instruction-following" capabilities of the system's core components. It is also convenient to transfer LongRAG to other domains by leveraging the pipeline and fine-tuning strategy.

Extensive performance comparisons and quantitative ablation studies conducted on three multi-hop datasets from LongBench (Bai et al., 2023b) demonstrate the superiority and effectiveness of LongRAG. The results suggest that LongRAG significantly outperformed both long-context LLMs and advanced RAG methods. We also discuss LongRAG's performance with different fine-tuned LLMs and confirm its strong robustness and transferability. To sum up, our contributions are summarized as follows: **1)** We construct LongRAG, a general, dual-perspective, and robust RAG system paradigm. It significantly surpasses long-context LLM (up by 6.94%), mainstream advanced RAG (up by 6.16%), and Vanilla RAG (up by 17.25%). **2)** We identify and address RAG's limitations in LCQA. We develop two plug-and-play components (i.e., Information Extractor and CoT-guided Filter) to explore global information and factual details, enhancing understanding of complex long contexts. **3)** We implement a novel automated fine-tuning data construction pipeline and a multi-task training strategy with multi-length long-context data. They facilitate the application of our paradigm to diverse specific-domain data in real-world scenarios.

2 Related Works

2.1 Long-Context LLMs

LLMs usually need to handle complex and long-context inputs in the real world. The context window length of LLMs is limited by their training

sequence length, and inputs exceeding this window may result in considerable performance degradation (Zhao et al., 2023; Jin et al., 2024). Thus, recent studies focus on scaling the limited context length of existing LLMs to accommodate tasks requiring long contexts, e.g., long-context question-answering. Methods for scaling the context length are categorized into two main types: 1) One is methods for training or fine-tuning with long contexts, such as RMT (Bulatov et al., 2022), Position Interpolation (Chen et al., 2023a), YaRN (Peng et al., 2023), Activation Beacon (Zhang et al., 2024a), LongLoRA (Chen et al., 2023b), LongRoPE (Ding et al., 2024), and LongAlign (Bai et al., 2024); 2) the other is non-fine-tuned methods include restricted attention-based approaches (Han et al., 2023; Xiao et al., 2023; Lu et al., 2024) and context compression methods (Jiang et al., 2023a; Li et al., 2023b). Generally, non-fine-tuned methods allow for plug-and-play and low-cost scaling LLMs. Fine-tuned methods typically show better performance but require higher training and data costs.

2.2 Retrieval-Augmented Generation

With the advent of the GPT era, RAG (Lewis et al., 2020; Guu et al., 2020) is regarded as a powerful technology for improving the response quality of LLMs (Izacard and Grave, 2021; Chung et al., 2022). RAG alleviates issues such as outdated and long-tail knowledge (He et al., 2023; Kandpal et al., 2023), hallucinations (Chen et al., 2023c; Zuccon et al., 2023), and lack of domain expertise (Li et al., 2023a; Shen et al., 2023) of LLMs by leveraging external knowledge, i.e., Wikipedia. Despite the success of RAG, its chunking strategy and direct incorporation of retrieved chunks into the generator result in incomplete information and substantial noise. Recently, advanced RAG models have been proposed to address these issues by filtering or re-ranking the retrieved knowledge to reduce noise (Yoran et al., 2023; Yan et al., 2024; Zhuang et al., 2023), designing a chunk-free strategy to mitigate semantic loss (Qian et al., 2024), and employing active retrieval to mine information (Asai et al., 2023; Jiang et al., 2023b).

2.3 Domain-Specific Fine-Tuning for RAG

Fine-tuning has gradually become a popular strategy (Ke et al., 2024) for enhancing the capabilities of components of RAG. Existing works include fine-tuning retrieval-related components to achieve better retrieval outcomes (Yan et al., 2024), fine-

tuning generators for more personalized outputs (Zhang et al., 2024b), and employing collaborative fine-tuning (Lin et al., 2023). Additionally, Zhou et al. (2023) discovered that fine-tuning LLMs with a limited quantity of high-quality data significantly enhances the performance of LLMs. This finding provides a robust theoretical basis for collaboratively fine-tuning multiple components within advanced RAG methodologies at a minimal data expense.

3 Preliminaries

3.1 Task Definition

Following the structure of Vanilla RAG (a retriever \mathcal{R} and a generator \mathcal{G}), the LongRAG system (cf., Figure 2) includes a **Long-Context Extractor** \mathcal{E} and a **CoT-guided Filter** \mathcal{F} after retrieval to extract global information I_g and identify factual details I_d . Specifically, given a question $q \in Q$ and a long-context corpus \mathcal{C} , \mathcal{R} receives a q and retrieves the top- k most relevant chunks $p_c \in P_c$. These p_c are obtained by segmenting source paragraphs $p \in P$. We then input p into \mathcal{E} , obtaining I_g , and p_c into \mathcal{F} to identify chunks containing factual details, defined as I_d , which are subsequently used by \mathcal{G} to generate a final answer to the question. It is worth noting that when discussing the system, P represents the source long-context paragraphs mapping from retrieved chunks P_c . However, when discussing fine-tuning instruction data \mathcal{D} , P denotes all corresponding paragraphs given for a question, including predefined supporting paragraphs P_s and given distracting paragraphs P_d .

3.2 Fine-Tuning Data Construction

To improve the "instruction following" ability of components and learn long-context styles, we craft a small but high-quality instruction-following dataset for supervised fine-tuning (SFT), named **LRGInstruction**, via ChatGLM3-32B-128k (Du et al., 2022; Zeng et al., 2023) as teacher LLM. We select the training sets of three complex English multi-hop datasets released by Trivedi et al. (2023) – HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MusiQue (Trivedi et al., 2022), as well as the English dataset QASPER with longer contexts (Dasigi et al., 2021), to jointly develop our LRGInstruction. Among them, QASPER with more lengthy contexts promotes LLMs to further learn the long-context style. The construction pipeline is automated, that is, you

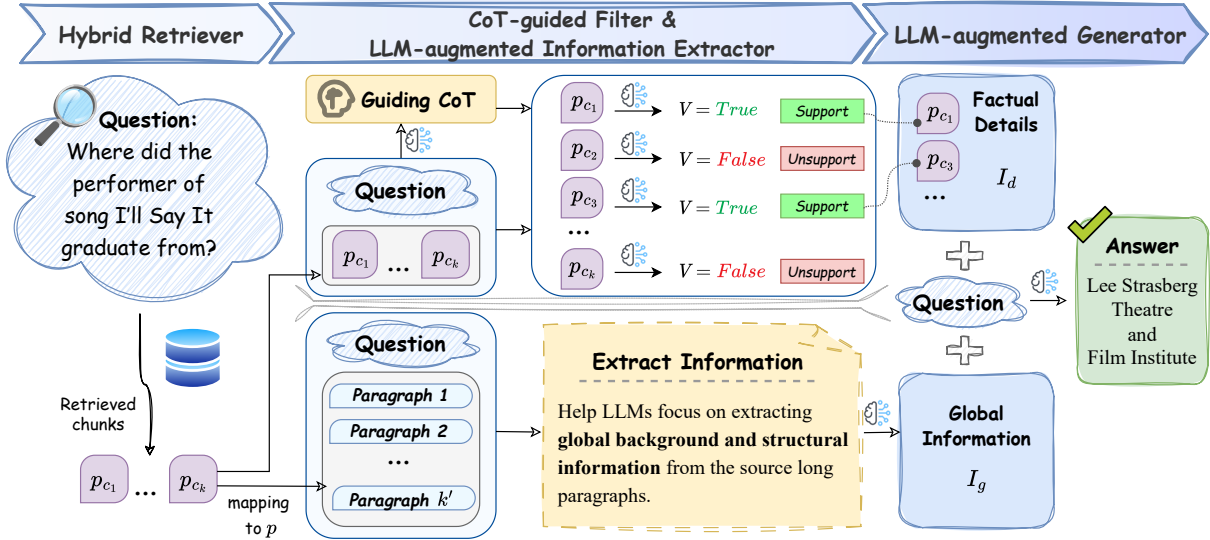


Figure 2: **An overview of LongRAG.** Our system involves four sub-components: Hybrid Retriever receives a question and retrieves the top- k most relevant chunks p_c ; CoT-guided Filter generates global key clues to analyze their relevance one by one, obtaining a set of "True" chunks as I_d ; Meanwhile, LLM-augmented Information Extractor sequentially maps p_c to the source long-context paragraph p to extract effective global information I_g ; LLM-augmented Generator promotes knowledge interaction between I_g and I_d to generate the final answer.

can automatically generate high-quality fine-tuning instruction data from any specific domain. In addition, the results of experiments indicate that we only need 2600 samples to fine-tune the LLMs used in components to achieve good performance in LCQA tasks. The construction pipeline is introduced as follows (**more details in Appendix C**).

Data Pre-Processing. To learn long-context style, we discard any question-answer pairs with insufficient context length (see details in Appendix C.1). Then, we keep all supporting paragraphs of questions P_s and randomly retain a subset of distracting paragraphs P_d . The random strategy is designed to simulate the distribution of the number of recalls executed in reality. To sum up, we define the elements of pre-processed dataset as follows: question $q \in Q$, multiple corresponding paragraphs $p \in P$, including supporting paragraphs P_s and distracting paragraphs P_d to the question, and answer $\alpha \in A$, mathematically $\langle Q, \{P_s \cup P_d\}, A \rangle$.

Long-Context Extractor Data. We fine-tune the long-context extractor to improve its capacity to extract global information from the source long paragraphs. First, we consider all P_s of each question as effective global information. These questions and their global information serve as input for zero-shot in-context learning (ICL) to gain global background and structure information, which act as golden outputs (see Appendix C.2 for details). Subsequently, to enhance the robustness of the pipeline,

we validate the efficacy of the golden outputs via an LLM-based self-evaluator and retain the golden outputs that are deemed valid.

CoT-guiding Data & Filtering Data. The training data for the CoT-guided filter component is constructed in two stages: the CoT guidance and the filtering stage. Key insights and clues for question resolution reside within P_s . Thus, for the CoT guidance stage, the LLM is expected to examine the semantic relations and factual details for question-solving within P_s to generate a guiding CoT. This process also employs a self-evaluator to evaluate the reliability of the CoT outputs as golden data. In the subsequent filtering stage, We merge q with a corresponding p and its guiding CoT as the gold data (see Appendix C.3 for details). P_s and P_d each account for half in P .

Task-Oriented Data. Question-answer pairs $\langle Q, A \rangle$ and P are already present in \mathcal{D} , and we simply need to reorganize their format.

4 The LongRAG System

4.1 Hybrid Retriever

The hybrid retriever begins with a given question and then recalls k chunks. Before the retrieval, the long-context p requires further segmentation into chunks p_c . Specifically, we impose a length limit on chunks, with sentences as the smallest division unit. We then employ a sliding window to

extend the context by adding overlapping content from the end of the previous sentence, preventing semantic disruption at truncation points. Short chunks at the end of p are merged with preceding chunks to ensure better semantic cohesion. Inspired by Re²G (Glass et al., 2022), we utilize a dual-encoder¹ structure for rapid retrieval at a coarse-grained level, and a cross-encoder² to capture the deep semantic interaction for further retrieval at a fine-grained level. The engineering implementation ensures efficient retrieval through the use of FAISS (Johnson et al., 2019).

4.2 LLM-augmented Information Extractor

In long-context QA with low evidence density, the complete evidence supporting answers is usually scattered across multiple locations. From a global perspective, this evidence not only contains its own knowledge but also implicitly stores logical and sequential connections among chunks. Retrieved chunks, truncated by fixed windows, struggle to carry additional global information. Furthermore, when the retrieved chunks originate from the same p , their order may be inconsistent with the original semantic order in p , resulting in providing disordered semantic information to downstream LLMs. To address these issues, we map the short-form chunks p_c back to their source long-context paragraphs p , using a mapping function $f_m(\cdot)$:

$$f_m(p_{c_1}, p_{c_2}, \dots, p_{c_k}) \rightarrow p_1, p_2, \dots, p_{k'} \quad (1)$$

where k and k' ($k \leq k'$) denote the number of pre-mapping p_c and post-mapping p , respectively. When multiple p_c correspond to the same p , we keep only the p corresponding to the p_c with the highest semantic similarity to the question q . This mapping strategy maximizes the recovery of the context of question-relevant source paragraphs. Then, we concatenate k' paragraphs p and feed them into the prompt (see Appendix D) of the LLM-augmented information extractor for employing zero-shot ICL.

$$I_g = \text{LLM}(\text{prompt}_e(q, p_1 || p_2 || \dots || p_{k'})) \quad (2)$$

The prompt template of the LLM-augmented information extractor, defined as $\text{prompt}_e(\cdot)$, guides the

¹We use E5-large model for dual-encoder: <https://huggingface.co/intfloat/multilingual-e5-large>

²We use mMiniLM as cross-encoder model: <https://huggingface.co/nreimers/mmarco-mMiniLMv2-L12-H384-v1>

LLM to ultimately obtaining global information I_g enriched with extensive long-context background and structural knowledge.

4.3 CoT-guided Filter

It is not always the case that retrieved chunks p_c will assist in answering questions, particularly in multi-hop questions that involve complex reasoning chains and long-context paragraphs with low evidence density. The retrieved chunks usually contain substantial redundancy; some of chunks can even be entirely redundant. This complexity makes it difficult to ascertain whether a chunk holds the key information for solving multi-hop questions.

To address this, we develop the CoT-guided filter with a two-stage strategy. The initial stage, CoT guidance, generates a CoT with a global perspective based on the retrieval semantic space, outlining the global clues for answering the question. Here's the mathematical expression of CoT-guidance stage:

$$\text{CoT} = \text{LLM}(\text{prompt}_c(q, p_{c_1} || \dots || p_{c_k})) \quad (3)$$

where k denotes the number of chunks p_c , and $\text{prompt}_c(\cdot)$ is the prompt template of yielding CoT based on LLMs. Subsequently, in the filtering stage, these CoTs serve as global clues, guiding LLMs step by step to focus on relevant knowledge throughout the reasoning chain. They equip filters with the ability to judge the relevance between questions and chunks using a high-dimensional perspective. This aids the system in inferring multi-hop semantic associations and meticulously examining all available factual details in contexts of low evidence density. Overall, this phase achieves high-quality identification of factual details and secures reliable relevance labels for question-chunk pairs. We use these labels to precisely filter irrelevant chunks p_c and avoid deleting crucial factual details, thus ensuring low redundancy input for the downstream generator.

$$V(q, p_c, \text{CoT}) = \begin{cases} \text{True}, & \text{if } \langle \text{support} \rangle \\ \text{False}, & \text{otherwise} \end{cases} \quad (4)$$

$$I_d = \{p_c \mid V(q, p_c, \text{CoT}) = \text{True}\}$$

Equation (4) describes the process of the filtering stage. $V(\cdot)$ returns a binary label to assess whether the chunk p_c supports answering the q according to the clues within the CoT. We iteratively assess each p_c via the function $V(\cdot)$. These chunks marked as "True" are considered as a set of chunks containing factual details information, defined as I_d .

Model	HotpotQA	2WikiMQA	MusiQue	Average
# Long-Context LLM Methods #				
LongAlign-7B-64k (<i>Llama2</i>) (Bai et al., 2024)	48.85	28.56	25.14	34.18
LongLoRA-13B-32k (<i>Llama2</i>) (Chen et al., 2023b)	47.45	42.92	29.46	39.94
# Advanced RAG Methods #				
CFIC-7B (<i>Llama2</i>) (Qian et al., 2024)	34.00	-	14.70	24.35
CRAG (<i>GPT-3.5-Turbo</i>) (Yan et al., 2024)	52.04	41.13	25.34	39.50
Self-RAG (<i>GPT-3.5-Turbo</i>) (Asai et al., 2023)	50.51	46.75	24.62	40.63
# RAG-Base (Vanilla RAG) #				
Vicuna-v1.5-7B-16k (Zheng et al., 2023)	38.63	27.92	15.68	27.41
Qwen-1.5-7B-32k (Bai et al., 2023a)	45.70	34.69	25.08	35.16
Llama3-8B-8k (Touvron et al., 2023)	48.25	43.47	19.66	37.13
ChatGLM3-6B-32k (Du et al., 2022)	52.57	42.56	25.51	40.21
GPT-3.5-Turbo-16k	50.17	45.32	21.84	39.11
GPT-3.5-Turbo	52.31	43.44	25.22	40.32
Llama3-70B-8k	52.33	50.23	25.49	42.68
GLM-4	57.41	52.91	27.55	45.96
# Ours with SFT #				
LongRAG-Llama2-7B-4k	53.85	45.61	26.22	41.89
LongRAG-Llama2-13B-4k	57.05	49.95	33.63	46.88
LongRAG-Qwen-1.5-7B-32k	52.91 (7.21↑)	46.65 (11.96↑)	31.85 (6.77↑)	43.80 (8.65↑)
LongRAG-Llama3-8B-8k	52.39 (4.14↑)	49.67 (6.20↑)	31.70 (12.04↑)	44.59 (7.46↑)
LongRAG-Vicuna-v1.5-7B-16k	55.55 (16.92↑)	50.13 (22.21↑)	28.29 (12.61↑)	44.66 (17.25↑)
LongRAG-ChatGLM3-6B-32k	55.93 (3.36↑)	54.85 (12.29↑)	33.00 (7.49↑)	47.93 (7.71↑)
# Ours without SFT #				
LongRAG-GPT-3.5-Turbo	56.17 (3.86↑)	51.37 (7.93↑)	32.83 (7.61↑)	46.79 (6.47↑)
LongRAG-GPT-3.5-Turbo-16k	59.11 (8.94↑)	51.25 (5.93↑)	30.37 (8.53↑)	46.91 (7.80↑)
LongRAG-GLM-4	62.11 (4.70↑)	57.16 (4.25↑)	38.40 (10.85↑)	52.56 (6.60↑)

Table 1: **Results (%) of overall performance on three multi-hop datasets.** The "Grey Areas" represent different categories of baselines or our system with different fine-tuning settings. "**Bold Font**" denotes the highest absolute value, while "Underlined Font" expresses the highest relative gain value compared to Vanilla RAG. Ours with (or without) SFT indicates we employ fine-tuned (or non-fine-tuned) LLMs in all LLM-augmented components. All model types are "chat". We calculate the increase in ours compared to Vanilla RAG, such as "17.25↑".

4.4 LLM-augmented Generator

Global information I_g encompasses both background and structural information within the long-context corpus, while factual details information I_d refers to the filtered chunk set with minimal noise and crucial evidence details. The generator boosts the interaction of knowledge across these two perspectives to produce answers α to questions. Here is the formula for the generator \mathcal{G} , where $prompt_g(\cdot)$ is the prompt template of generator:

$$\alpha = \text{LLM}(prompt_g(I_g, I_d)) \quad (5)$$

4.5 Instruction-Tuning

We adopt a collection of industry-leading models as our foundational LLMs: ChatGLM (Du et al., 2022; Zeng et al., 2022), Qwen1.5 (Bai et al., 2023a), Vicuna (Zheng et al., 2023), Llama2, and Llama3 (Touvron et al., 2023). They are all open-source and support multi-lingual, multi-tasking. We have fine-tuned them using 2,600 high-quality data sourced from LRGInstruction. Specifically, we employ all four types of data in LRGInstruction collectively to train a model that is used in the extractor, the filter, and the generator. Furthermore, this data has undergone length filtering and has

been standardized into a QA instruction style. During training, all models utilize the Llama-factory library and 8xA100 GPUs (80G each), employing training methods with DeepSpeed+ZeRO3+CPU offloading+flash attention strategies (Rasley et al., 2020; Dao et al., 2022). The training parameters are set with a batch size of 8, a gradient accumulation step of 12, and 3 epochs (totaling 81 steps).

5 Experiment

5.1 Experimental Setup

Datasets & Evaluation. We select three challenging multi-hop datasets – HotpotQA, 2WikiMultiHopQA (2WikiMQA), and MusiQue – from the Longbench (Bai et al., 2023b) for evaluation, rather than using raw datasets. We standardize these data to adapt to RAG tasks (**more details in Appendix B.2**), and report the F1-score as evaluation metrics for all three datasets. Statistics of experimental datasets are shown in Table 2.

Baselines & LLMs. To validate the superiority of our LongRAG in multiple dimensions, we utilize three categories of baselines: 1) Long-Context LLM Methods – **LongAlign** (Bai et al., 2024) and **LongLoRA** (Chen et al., 2023b); 2)

Dataset	HotpotQA	2WikiMQA	MuSiQue
Num of Samples	200	200	200
Avg. Length of p	1092	535	1032
Num of p	1715	1464	1877
Avg. Length of P	9151	4887	11214

Table 2: **Statistics of experimental data.** "Avg. Length" stands for the average word count.

Advanced RAG Methods – **CFIC** (Qian et al., 2024), **CRAG** (Yan et al., 2024), and **Self-RAG** (Asai et al., 2023); 3) **Vanilla RAG** (only retriever \mathcal{R} and generator \mathcal{G}) based on various LLMs. These LLMs range from small parameter-size (6b~8b) models like ChatGLM3-6B-32k (Du et al., 2022), Qwen1.5-7b-32k (Bai et al., 2023a), Vicuna-v1.5-7b-16k (Zheng et al., 2023), and Llama3-8B-8k (Touvron et al., 2023) to large parameter-size online models like GPT-3.5-Turbo³ (gpt-3.5-turbo-0125) and GLM-4⁴ (glm-4).

Others. In our experiments, all token lengths are measured by ChatGLM tokenizer. We evaluate four different retrieval strategies to analyze the performance of LongRAG comprehensively (more details and results in Appendix A.1). Specifically, we represent four retrieval strategies as "chunk size*top- k ", including "200*7", "200*12", "500*3", and "500*5". By default, we set the chunk size to 200 words and the top- k value to 7.

5.2 Overall Performance

In this section, we perform a multi-dimensional comparison and analysis of the overall performance results in Table 1.

Ours vs. Long-Context LLM Methods. We align the parameter size of *Llama2* and compare LongRAG with the results of LongAlign and LongLoRA. Our system paradigm using SFT achieves the highest performance on all datasets. In addition, we also observe that the LongRAG system paradigm equipping other similar parameter-size LLMs consistently surpasses baselines within Long-context LLM methods across all datasets. These achievements confirm the superiority of our system across all datasets. This occurs because long-context LLMs often overlook crucial factual details in the middle, while LongRAG precisely and robustly perceives factual details. Overall, our system serves as a more effective technical solution for LCQA.

³<https://openai.com/blog/chatgpt>

⁴Due to resource limitations, we perform the API of glm4 with an 8k token window. <https://open.bigmodel.cn>.

Ours vs. Other RAG. We compare LongRAG with two categories of RAG baselines, advanced RAG and Vanilla RAG (RAG-Base, R&B). We employ the LangGraph library⁵, integrated within the LangChain framework, to reproduce Self-RAG and CRAG. **First**, compared to the advanced RAG, especially Self-RAG, our LongRAG achieves a 6.16% improvement across three datasets on average. This is due to the self-reflective chain decision-making in Self-RAG, which can, in certain cases, amplify decision errors, leading to catastrophic loss of factual details. Similarly, CRAG exhibits non-robust evaluation behaviors, making it challenging to handle complex, multi-hop long-context questions. **Second**, compared to the R&B, all LLMs applied in our system exhibit significant improvements (up to 17.25%). Vanilla RAG segments long contexts into smaller semantic units, hindering the downstream generator from accessing a more coherent long-context background and the original long-context structure. Based on the above analysis, our system, after performing extractor and filter, acquires higher-quality and less noise knowledge, thus generating more accurate answers.

Small-Size vs. Large-Size LLMs. We find that the LongRAG system paradigm, whether employing fine-tuned small-size or non-fine-tuned large-size LLMs, consistently outperforms other baseline methods across all datasets. Most importantly, LongRAG using the fine-tuned ChatGLM3-6B-32k achieves better performance than using non-fine-tuned GPT-3.5-Turbo. These results prove our system paradigm boosts the ability to analyze and process complex long contexts, as well as "instruction following" capability. It also compensates for the limitations observed in small-size LLMs, particularly in long-context in-context learning (ICL) and understanding complex information.

5.3 Ablation Study

The ablation study (Table 3) reports results within five strategies to highlight the effectiveness of the information extractor and CoT-guided filter. In the following paragraphs, we explore the reasons for the performance gains.

RAG-Long vs. RAG-Base. RAG-Long (R&L) refers to mapping the p_c back to the p and then directly putting a set of p into the generator to output a response. The R&L strategy fails to robustly achieve performance improvements over

⁵<https://github.com/langchain-ai/langgraph>

Model	HotpotQA					2WikiMQA					MusiQue				
	R&B	R&L	Ext.	Fil.	E&F	R&B	R&L	Ext.	Fil.	E&F	R&B	R&L	Ext.	Fil.	E&F
# Ours with SFT #															
LongRAG-ChatGLM3-6B-32k	51.48	54.00	<u>55.11</u>	49.01	55.93	46.61	44.83	<u>52.53</u>	48.83	54.85	24.02	33.15	32.98	27.70	<u>33.00</u>
LongRAG-Qwen1.5-7B-32k	47.09	48.93	<u>50.01</u>	49.11	52.91	35.78	37.72	<u>42.91</u>	38.98	46.65	20.68	26.08	<u>29.60</u>	23.67	31.85
LongRAG-Vicuna-v1.5-7B-16k	51.63	50.18	55.94	52.34	<u>55.55</u>	39.45	43.53	<u>49.57</u>	41.18	50.13	25.30	25.28	<u>29.25</u>	29.29	28.29
LongRAG-Llama3-8B-8k	49.45	50.49	<u>51.77</u>	49.64	52.39	39.79	37.16	<u>46.80</u>	42.40	49.67	21.41	22.90	33.85	23.47	<u>31.70</u>
# Ours without SFT #															
LongRAG-ChatGLM3-6B-32k	<u>52.57</u>	50.19	52.27	53.36	52.07	42.56	42.92	<u>44.95</u>	42.94	46.08	25.51	29.93	28.27	23.99	<u>28.45</u>
LongRAG-Qwen1.5-7B-32k	45.70	49.72	<u>50.74</u>	45.70	50.80	34.69	<u>35.49</u>	39.53	34.69	39.53	<u>25.08</u>	25.85	29.75	<u>25.08</u>	29.75
LongRAG-Vicuna-v1.5-7B-16k	38.63	30.40	<u>41.45</u>	39.46	43.18	27.92	20.68	<u>29.08</u>	29.89	30.85	15.68	8.92	17.65	<u>16.35</u>	16.98
LongRAG-Llama3-8B-8k	48.25	48.72	52.44	47.75	52.19	43.47	41.59	47.34	42.22	<u>46.57</u>	19.66	23.62	24.90	20.06	24.99
LongRAG-GPT-3.5-Turbo	52.31	55.30	<u>56.15</u>	50.90	56.17	43.44	45.03	53.29	39.49	<u>51.37</u>	25.22	28.65	<u>32.17</u>	24.41	32.83
LongRAG-GPT-3.5-Turbo-16k	50.17	49.80	60.06	47.10	59.11	45.32	46.80	51.26	46.38	<u>51.25</u>	21.84	25.09	<u>26.92</u>	22.02	30.37
LongRAG-GLM-4	57.41	56.17	<u>61.07</u>	55.41	62.11	52.91	48.98	<u>54.22</u>	52.61	57.16	27.55	27.85	38.54	28.12	<u>38.40</u>

Table 3: **Results (%) of the ablation study.** We compare five strategies in two dimensions: with and without SFT. We highlight the highest ("**Bold Font**") and second-highest ("_") results per model. R&B, R&L, Ext., Fil., and E&F represent RAG-Base, RAG-Long, Extractor, Filter, and Extractor & Filter, respectively.

R&B. Specifically, the R&L strategy feeds the continuous long-context space into the LLM, unlike the R&B disrupts the semantic continuity of long contexts. Therefore, R&L enables to capture of a broader continuity of the source semantic space; however, it also risks introducing excessive noise.

Extractor vs. RAG-Long. The extractor builds upon the R&L to effectively extract pertinent long-context information. Specifically, the extractor strategy refers to the system first extracting global information I_g from the mapped source long paragraphs, and then using I_g as supplementary input alongside retrieved chunks p_c to the generator to enhance answer quality. The system using the extractor strategy presents substantial improvements across all three datasets, particularly on larger-size LLMs that exhibit stronger in-context learning capability. This improvement stems from recognizing the challenge of directly deriving answers from lengthy contexts; therefore, we first leverage the LLMs' capability to extract global structures and background knowledge as supplements for generating the final answer. The extractor strategy effectively mitigates the issue of low-quality responses in the R&L strategy caused by directly feeding redundant long passages into LLMs, while also providing LLMs with additional and concise global structure and contextual relationship information. Additionally, in most instances, the extractor is the primary contributor to performance gains, second only to the joint strategy, Extractor & Filter (**E&F**).

Filter vs. RAG-Base. Using the filter alone based on R&B improves the performance only marginally in a few cases. This occurs because filtering is, after all, a process of information reduction. Therefore, it can only display markedly performance when

used in conjunction with the Extractor.

Extractor & Filter vs. Others. E&F serves as a joint strategy with two pluggable components within the RAG system, achieving the best performance in the majority of cases. It outperforms the R&L strategy by providing refined information with less noise, thereby effectively alleviating the "*lost in the middle*" issue. Specifically, the role of the Extractor is to capture globally effective information from long contexts, while the Filter flexibly selects factual details through interactions between the question and relevant paragraphs. Results suggest employing both E&F components yields a more helpful and concise set of information compared to using a single component. However, it is worth mentioning that a minority of cases where E&F underperforms compared to Extractor alone do not imply that the Filter is ineffective. In fact, when the built-in LLM possesses strong "*instruction-following*" capabilities (e.g., GLM-4 and fine-tuned small-size LLMs), adding the Filter is more likely to boost system performance. Plus, the Filter can reduce the number of tokens input into downstream LLMs. From the results in Table 3 and Figure 3, it is evident that using the Filter can save token costs during the generation phase while achieving performance comparable to or even better than using the Extractor alone. Furthermore, we find that not all researchers can afford the high costs of powerful API LLMs (e.g., GPT-3.5-Turbo). Our method offers an alternative by using more affordable open-source local LLMs for components before the generator, instead of relying on expensive online APIs throughout the entire inference process. Therefore, if the goal is to balance performance and cost, E&F is crucial.

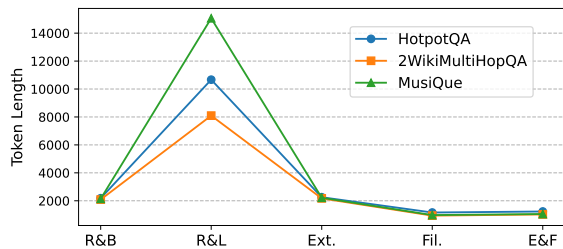


Figure 3: Trends of token lengths fed into the Generator \mathcal{G} of five component strategies on three datasets.

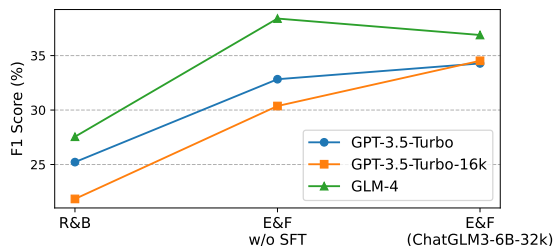


Figure 4: Analysis of the transferability of Extractor&Filter on dataset MusiQue.

5.4 Discussion

Analysis of Token Length Trends. Figure 3 illustrates the token lengths inputted into the generator \mathcal{G} for all datasets after undergoing the five strategies. The results indicate a consistent trend across all datasets. Specifically, our E&F strategy feeds \mathcal{G} fewer tokens but achieves superior outcomes, however, R&L feeds the most without corresponding systematic gains, which indicates we can obtain higher quality information through E&F.

Component Transferability. As shown in Figure 4, E&F (ChatGLM3-6B-32k) means we employ ChatGLM3-6B-32k as the built-in LLM of extractor \mathcal{E} and filter \mathcal{F} , while the generator \mathcal{G} uses other powerful online LLMs, e.g., GPT-3.5-Turbo. E&F w/o SFT represents the same meanings in Table 3, that is, we apply the same built-in LLM for the \mathcal{E} , \mathcal{F} , and \mathcal{G} . Results reveal we transfer the expensive powerful online LLMs of \mathcal{E} and \mathcal{F} to a low-cost local model while achieving excellent results. It can surpass GPT-3.5-Turbo and rival the GLM-4.

6 Conclusion

We build an effective and robust RAG system paradigm — **LongRAG** — which enhances RAG’s performance in LCQA tasks via a dual information perspective. LongRAG addresses two main issues faced by existing methods: 1) the incomplete collection of long-context information; and 2) the dif-

ficulty in precisely identifying factual information amid substantial noise. We conduct extensive multi-dimensional experiments, which demonstrate the superiority of LongRAG and the effectiveness of our proposed components and fine-tuning strategy. LongRAG significantly outperforms long-context LLMs, advanced RAG methods, and Vanilla RAG based on various LLMs. Our plug-and-play components successfully use small parameter-size LLMs, replacing expensive online API resources with low-cost local deployment solutions, while better than GPT-3.5-Turbo. Additionally, we provide an automated pipeline for fine-tuning instruction data construction, which greatly facilitates the application of our system to other specific-domain data.

7 Limitations

This paper presents a general-purpose and corpus-level retrieval-augmented generation system paradigm for long-context question answering, termed LongRAG. While the system paradigm brings significant advancements and proves effective, it is also subject to certain limitations that merit discussion.

One-time Retrieval Dependency. In this study, we only investigated the performance of the information extractor and CoT-guided filter in a one-time retrieval scenario. The quality of CoTs and source documents for answering depends on the quality of single-pass retrieved chunks. Consequently, low-quality one-time retrieval can indirectly undermine the effectiveness of our core components. Moving forward, we anticipate that an effective avenue of improvement could develop an adaptive multi-round retrieval strategy through interaction with core components.

Dataset Annotation Bias. Although we have used the 32-billion parameter ChatGLM3 model to generate high-quality fine-tuning datasets, models of this scale may still be susceptible to annotation biases inherent in self-generated datasets. Such biases could impair the contextual understanding of the fine-tuned models across diverse tasks and domains, potentially undermining the overall system performance. It is therefore valuable to thoroughly investigate the performance of instruction datasets created by LLMs of various scales in cross-domain and multi-task environments.

Acknowledgments

This work is supported by the Natural Science Foundation of China (NSFC) 62276148 and 62425601, Tsinghua University (Department of Computer Science and Technology) -Siemens Ltd., China Joint Research Center for Industrial Intelligence and Internet of Things (JCIOT) and New Cornerstone Science Foundation through the XPLOER PRIZE.

References

- Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. [Gemini: A family of highly capable multimodal models](#). *CoRR*, abs/2312.11805.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#). *CoRR*, abs/2310.11511.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023a. [Qwen technical report](#). *CoRR*, abs/2309.16609.
- Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024. [Longalign: A recipe for long context alignment of large language models](#). *CoRR*, abs/2401.18058.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023b. [Longbench: A bilingual, multitask benchmark for long context understanding](#). *arXiv preprint arXiv:2308.14508*.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. 2022. [Recurrent memory transformer](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Avi Caciularu, Ido Dagan, Jacob Goldberger, and Arman Cohan. 2022. [Long context question answering via supervised contrastive learning](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2872–2879. Association for Computational Linguistics.
- Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. [Benchmarking large language models in retrieval-augmented generation](#). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 17754–17762. AAAI Press.
- Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. [Extending context window of large language models via positional interpolation](#). *CoRR*, abs/2306.15595.
- Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023b. [Longlora: Efficient fine-tuning of long-context large language models](#). *CoRR*, abs/2309.12307.
- Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023c. [Hallucination detection: Robustly discerning reliable answers in large language models](#). In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 245–255. ACM.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang,

- Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. [Scaling instruction-finetuned language models](#). *CoRR*, abs/2210.11416.
- Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. [Flashattention: Fast and memory-efficient exact attention with io-awareness](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. [A dataset of information-seeking questions and answers anchored in research papers](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4599–4610. Association for Computational Linguistics.
- Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. [Longrope: Extending LLM context window beyond 2 million tokens](#). *CoRR*, abs/2402.13753.
- Zican Dong, Tianyi Tang, Junyi Li, and Wayne Xin Zhao. 2023. [A survey on long text modeling with transformers](#). *CoRR*, abs/2302.14502.
- Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. [GLM: general language model pretraining with autoregressive blank infilling](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 320–335. Association for Computational Linguistics.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. [Retrieval-augmented generation for large language models: A survey](#). *CoRR*, abs/2312.10997.
- Michael R. Glass, Gaetano Rossiello, Md. Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. [Re2g: Retrieve, rerank, generate](#). *CoRR*, abs/2207.06300.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. [Retrieval augmented language model pre-training](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.
- Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2023. [Lm-infinite: Simple on-the-fly length generalization for large language models](#). *CoRR*, abs/2308.16137.
- Hangfeng He, Hongming Zhang, and Dan Roth. 2023. [Rethinking with retrieval: Faithful large language model inference](#). *CoRR*, abs/2301.00303.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.
- Gautier Izacard and Edouard Grave. 2021. [Distilling knowledge from reader to retriever for question answering](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023a. [Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression](#). *CoRR*, abs/2310.06839.
- Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. [Active retrieval augmented generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7969–7992. Association for Computational Linguistics.
- Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. [LLM maybe longlm: Self-extend LLM context window without tuning](#). *CoRR*, abs/2401.01325.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. [Billion-scale similarity search with GPUs](#). *IEEE Transactions on Big Data*, 7(3):535–547.
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. [Large language models struggle to learn long-tail knowledge](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.
- Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. [Bridging the preference gap between retrievers and llms](#). *CoRR*, abs/2401.06954.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. [Retrieval-augmented generation](#)

- for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Xianzhi Li, Samuel Chan, Xiaodan Zhu, Yulong Pei, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. 2023a. [Are chatgpt and GPT-4 general-purpose solvers for financial text analytics? A study on several typical tasks](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP 2023 - Industry Track, Singapore, December 6-10, 2023*, pages 408–422. Association for Computational Linguistics.
- Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023b. [Compressing context to enhance inference efficiency of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6342–6353. Association for Computational Linguistics.
- Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Scott Yih. 2023. [RA-DIT: retrieval-augmented dual instruction tuning](#). *CoRR*, abs/2310.01352.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Trans. Assoc. Comput. Linguistics*, 12:157–173.
- Yi Lu, Xin Zhou, Wei He, Jun Zhao, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. [Longheads: Multi-head attention is secretly a long context processor](#). *CoRR*, abs/2402.10685.
- Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. [Yarn: Efficient context window extension of large language models](#). *CoRR*, abs/2309.00071.
- Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. 2024. [Grounding language model with chunking-free in-context retrieval](#). *CoRR*, abs/2402.09760.
- Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. [Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters](#). In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3505–3506. ACM.
- Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. [In chatgpt we trust? measuring and characterizing the reliability of chatgpt](#). *CoRR*, abs/2304.08979.
- Ravi Theja. 2023. [Evaluating the ideal chunk size for a rag system using llamaindex](#).
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#). *CoRR*, abs/2302.13971.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [Musique: Multi-hop questions via single-hop question composition](#). *Trans. Assoc. Comput. Linguistics*, 10:539–554.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. [Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. [Efficient streaming language models with attention sinks](#). *CoRR*, abs/2309.17453.
- Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. [Corrective retrieval augmented generation](#). *CoRR*, abs/2401.15884.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [Hotpotqa: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. [React: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. [Making retrieval-augmented language models robust to irrelevant context](#). *CoRR*, abs/2310.01558.
- Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma,

- Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. [GLM-130B: an open bilingual pre-trained model](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Peng Zhang, Yuxiao Dong, and Jie Tang. 2022. [GLM-130B: an open bilingual pre-trained model](#). *CoRR*, abs/2210.02414.
- Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2024a. [Soaring from 4k to 400k: Extending llm’s context with activation beacon](#). *CoRR*, abs/2401.03462.
- Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024b. [RAFT: adapting language model to domain specific RAG](#). *CoRR*, abs/2403.10131.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. [Siren’s song in the AI ocean: A survey on hallucination in large language models](#). *CoRR*, abs/2309.01219.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. [A survey of large language models](#). *CoRR*, abs/2303.18223.
- 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](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. [LIMA: less is more for alignment](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. [Open-source large language models are strong zero-shot query likelihood models for document ranking](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 8807–8817. Association for Computational Linguistics.
- Guido Zuccon, Bevan Koopman, and Razia Shaik. 2023. [Chatgpt hallucinates when attributing answers](#). In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2023, Beijing, China, November 26-28, 2023*, pages 46–51. ACM.

A Additional Experimental Results

A.1 Results of Different Retrieval Strategies

Table 6, Table 7, and Table 8 display all of overall performance results. We evaluate four different retrieval strategies to analyze the performance of LongRAG comprehensively. These strategies include 200*7, 200*12, 500*3, and 500*5. For example, "200*7" stands for "chunk size*top- k ". By comparing these retrieval strategies, we observe that an intermediate value for the top- k setting tends to yield superior performance. This phenomenon arises from the extractor’s utilization of the source long paragraphs mapped from top- k recalled chunks. Too few recalled chunks may result in insufficient collection of extensive contextual information, while an excessive number may introduce more noise. Contrasting the outcomes of 200*7 and 500*3, we notice that, under comparable context length, a smaller chunk size coupled with a higher top- k recall number can maximize the acquisition of global information within the corpus space, thereby exhibiting enhanced performance. These results confirm the efficacy of the core components (\mathcal{E} and \mathcal{F}) in our system.

A.2 Component Transferability

We provide specific values in Figure 4 in section 5.4 with experimental results (Table 9, Table 10 and Table 11) for all datasets, including HotpotQA, 2WikiMultiHopQA, and MusiQue. In 2WikiMultiHopQA and HotpotQA, our system also exhibits component transferability similar to that in MusiQue. We conducted all experiments using ChatGLM3-6B-32k with SFT as a relatively low-cost local model.

A.3 Analysis of Token Length Trends

Figure 3 only shows the token length trend using ChatGLM3-6B-32k with SFT across five strategies. The specific values and the results of using more built-in fine-tuned LLMs are shown in Table 12, Table 13, and Table 14.

A.4 Additional Baseline Results

As an agent framework, ReAct can also be instantiated as an efficient RAG system based on adaptive retrieval (Yao et al., 2023). ReAct can answer questions through the process of "Thought/Action/Observation". In our experiment, we define "Action" as the retrieval action, meaning

Datasets	ReAct (<i>GPT-3.5-Turbo</i>)
HotpotQA	49.60
2WikiMultihopQA	41.86
MusiQue	27.81
Average	39.76

Table 4: Results of ReAct.

that when knowledge needs to be retrieved, the relevant information is retrieved from our local corpus \mathcal{C} . We have aligned the experimental parameters, and the results of the ReAct experiment are presented in Table 4.

B Experimental Details Explanation

B.1 Details of Baseline Replication

Self-RAG, CRAG, LongLoRA, and LongAlign produce too long responses, making it challenging to fairly compare them with our method using the F1-score as an evaluation metric. In other words, the long outputs result in lower scores for these baselines. Therefore, we select the LLM with a strong ability of "instruction-following", such as GPT-3.5-Turbo, and perform few-shot ICL on their outputs to produce the final answers. In the following paragraphs, we will introduce the specific experimental details involved in reproducing the results for Self-RAG and CRAG.

We employ the LangGraph library, integrated within the LangChain framework, to reproduce Self-RAG and CRAG. Specifically, Self-RAG employs an adaptive retrieval based on self-reflection. If the LLM identifies the retrieved chunks as irrelevant, or the generated outputs are regarded as unanswerable, Self-RAG will restart the search and answer process until the maximum number of rounds. In our experiments, we set the maximum number of retrieval rounds to 3. If, upon reaching this round limit, all retrieved documents are still considered irrelevant, there are two answer strategies: The first strategy uses all chunks retrieved during the final round, while the second strategy involves answering without using the retrieved chunks. In Table 1 of the main paper, we report the results of the first strategy, which shows higher results than those of the second strategy. Additionally, we present the performance of the second strategy in Table 5.

CRAG has implemented a fallback strategy to prevent a steep decline in response quality due to all retrieved chunks being filtered out. When the retrieved chunks are considered insufficient to an-

Datasets	Self-RAG (<i>GPT-3.5-Turbo</i>)
HotpotQA	44.99
2WikiMultihopQA	19.79
MuSiQue	23.49

Table 5: Results of Self-RAG via the second strategy.

swer the question, it is supplemented with external knowledge retrieved from the web. For a fair reproduction in our experiments, when faced with similar issues, we rewrite the question and conduct another retrieval from our corpus \mathcal{C} . Since our corpus contains all the relevant information necessary to answer the question, we do not need to retrieve external knowledge from the web.

B.2 Details of the Corpus

Our experimental datasets and the corpus used for knowledge retrieval are constructed based on LongBench. The multi-hop QA datasets of LongBench include questions, answers, and multiple corresponding paragraphs concatenated to form long contexts of each question. To adapt it for the RAG system, we split long contexts into individual corresponding paragraphs. Since each paragraph is a semantically coherent and complete Wikipedia paragraph, we treat each paragraph p as an independent knowledge unit. After deduplication, the paragraphs from all questions form the corpus \mathcal{C} .

C Details of LRGInstruction

We construct an instruction dataset for fine-tuning, comprising four types of data, each designed to enhance the "instruction-following" capability of corresponding components. The four types of data include **Long-Context Extractor data**, **CoT-guiding Data**, **Filtering Data**, and **Task-Oriented Data**. To be specific, long-context extractor data is utilized to enhance the capabilities of the LLM-augmented extractor. CoT-guiding data and filtering data are applied to strengthen the abilities of the two-stage CoT-guided filter. Question and answer data are utilized to enhance the generator’s capability, learning the specific answering style required for tasks. We present examples of all the pipelines used for data construction and formats of the generated data (golden data) in Table 16, Table 17 and Table 18. Specific examples of four types of golden data are also shown in Table 19, Table 20, Table 21 and Table 22. To clearly distinguish between prompts for data construction and

generated instruction data, we mark prompts in each pipeline as [STEP] and instruction data as [RESULT]. The following paragraphs will elaborate on the construction details and pipelines.

C.1 Data Pre-Processing

We further detail the random strategy. The number of distracting paragraphs P_d in our instruction data is randomly chosen within a specific range, from two up to the total length of P_d , mathematically expressed as $[2, \maxLen(P_d)]$. Moreover, we further detail how to discard any question-answer pairs with insufficient context length. Here, "insufficient context length" means that the total token length of all corresponding paragraphs provided for a question is lower than a specific threshold. Specifically, we use a threshold of 1.5k for HotpotQA and 2Wiki-MultiHopQA, and 2.5k for MusiQue. During the experiment, we find that this threshold setting preserves long-context samples, enabling the model to learn long-context styles and retain sufficient data for training. For QASPER, we do not filter any samples because the papers are inherently long.

C.2 Long-Context Extractor Data

In the construction pipeline (Table 16) for LLM-augmented extractor data, we aim to feed the question and P_s into the LLM, which outputs all the relevant information for answering the question. We provide the specific construction process and details shown in Table 16. We construct the initial dataset via [STEP-1], which global information as gold outputs. If the response of [STEP-1] is particularly short, we discard it due to a small amount of effective information, with a discard threshold of 20 tokens. Subsequently, in [STEP-2], we perform a self-evaluator of the gold output after [STEP-1]. Only samples that pass the validation (i.e., those for which the output in [STEP-2] is "True") are included in the final instruction dataset. The final [RESULT] presents the ultimate gold data (long-context extractor data) in this pipeline, and "{content}" represents P including both P_s and selected P_d by random strategy. This type of data enhances the LLM-augmented extractor to identify valuable evidence information from substantial lengthy context source paragraphs.

C.3 CoT-guiding Data & Filtering Data

In the CoT-guided filter, we employ a two-stage strategy to precisely and flexibly screen problem-related chunks while discarding redundant

chunks. The two types of data, CoT-guiding data ([**RESULT-1**]) and filtering data ([**RESULT-2**]) aim to enhance the "instruction-following" ability of the two-stage components of the CoT-guided filter, and better identify factual details. This construction pipeline and final constructed data are shown in Table 17. First, in [**STEP-1**], we generate a guiding CoT by inputting the question and all corresponding P_s . The generated CoT provides global clues for question-answering by performing in-context learning in all retrieved chunks. If the CoT is particularly short, we consider it a low-quality clue and discard it, with a discard threshold of 20 tokens. In [**STEP-2**], we then perform a self-evaluator of the guiding CoT [**STEP-1**] to verify the feasibility of the CoT in responding to the question. In the self-evaluator, we use the answers from the raw dataset as the basis for judging the quality of CoT. [**RESULT-1**] displays the instruction data constructed for the CoT-guided stage, named CoT-guiding data, and "{content}" represents P including both P_s and selected P_d by random strategy. Finally, for the filtering stage, we treat each paragraph p as a unit and regard given binary discrete labels in the raw dataset as gold labels, expressed as {status} in [**RESULT-2**]. The filtering stage instruction data is shown in [**RESULT-2**]. Its "{content}" represents each paragraph $p \in P$. It is worth noting that in the original dataset, the number of p marked as "True" is much lower than "False". To ensure the uniformity of the distribution, we select 100 samples with a status of "True" and 100 samples with a status of "False".

C.4 Task-Oriented Data

The questions and answers are already provided in the original datasets. We standardize their format to construct the question-answering data (see Table 18) in our fine-tuning instruction dataset. The "{content}" in [**RESULT**] represents P including both P_s and selected P_d by random strategy.

C.5 Statistics of LRGInstruction

To sum up, we derive four types of data from the training sets of the HotpotQA, 2WikiMultiHopQA, and MusiQue datasets, with each type of data containing 200 samples. This results in 800 samples per dataset and a total of 2400 samples across the three datasets. The token length of each instruction data is less than 7k. Furthermore, to adapt our RAG system to long-context QA, we also derive two types of data (i.e., long-context extractor data

and CoT-guiding data) using the QASPER dataset, each type of data with 100 samples, and each instruction data length ranging from 6k-29k. We list the statistics of our fine-tuning instruction dataset in Table 15.

D Prompts of LongRAG System

We present all prompts in LongRAG's components in Table 23. The "{content}" in different prompts represent different contextual information. To be specific, the "{content}" in the prompt of LLM-augmented information extraction represents all source long-context paragraphs p after the mapping strategy. In the prompt of the CoT guidance stage in the CoT-guided filter, it represents all retrieval chunks p_c , while in the prompt of the filtering stage, it represents each p_c .

E Answer Examples

We provide answer examples shown in Table 24, Table 25, and Table 26. LongRAG addresses the issues of incomplete information and "lost in the middle" found in Vanilla RAG and RAG-Long, while requiring fewer tokens inputted into the generator yet showing superior response performance.

Model	HotpotQA			
	200*7	200*12	500*3	500*5
# RAG Base (Vanilla RAG) #				
ChatGLM3-6B-32k	52.57	53.10	47.72	51.17
Qwen1.5-7B-32k	45.70	49.20	44.43	44.16
Vicuna-v1.5-7B-16k	38.63	34.35	37.23	35.32
Llama3-8B-8k	48.25	51.69	47.12	50.88
GPT-3.5-Turbo	52.31	55.21	52.84	51.21
GPT-3.5-Turbo-16k	50.17	53.58	48.02	48.84
Llama3-70B-8k	52.33	53.53	49.51	51.38
GLM-4	57.41	59.55	53.71	58.45
# Ours with SFT #				
LongRAG-ChatGLM3-6B-32k	55.93	54.36	50.72	54.67
LongRAG-Qwen1.5-7B-32k	52.91	52.27	49.70	50.69
LongRAG-Vicuna-v1.5-7B-16k	55.55	54.79	52.26	52.89
LongRAG-Llama3-8B-8k	52.39	52.00	49.05	54.62
# Ours without SFT #				
LongRAG-GPT-3.5-Turbo	56.17	56.06	55.63	55.11
LongRAG-GPT-3.5-Turbo-16k	59.11	51.55	48.45	55.57
LongRAG-GLM-4	62.11	60.55	55.36	61.14

Table 6: Overall performance of our LongRAG on HotpotQA dataset.

Model	2WikiMultiHopQA			
	200*7	200*12	500*3	500*5
# RAG Base (Vanilla RAG) #				
ChatGLM3-6B-32k	42.56	38.71	40.65	42.34
Qwen1.5-7B-32k	34.69	34.79	34.47	35.24
Vicuna-v1.5-7B-16k	27.92	26.39	32.76	26.36
Llama3-8B-8k	43.47	40.01	30.48	41.44
GPT-3.5-Turbo	43.44	40.06	43.17	39.69
GPT-3.5-Turbo-16k	45.32	39.09	43.31	42.49
Llama3-70B-8k	50.23	48.91	46.61	50.10
GLM-4	52.91	52.37	49.48	51.06
# Ours with SFT #				
LongRAG-ChatGLM3-6B-32k	54.85	58.51	49.28	53.51
LongRAG-Qwen1.5-7B-32k	46.65	45.23	42.96	44.55
LongRAG-Vicuna-v1.5-7B-16k	50.13	50.93	47.45	48.02
LongRAG-Llama3-8B-8k	49.67	51.41	43.80	49.70
# Ours without SFT #				
LongRAG-GPT-3.5-Turbo	51.37	56.55	48.16	48.60
LongRAG-GPT-3.5-Turbo-16k	51.25	45.45	44.08	44.21
LongRAG-GLM-4	57.16	52.90	44.93	50.05

Table 7: Overall performance of our LongRAG on 2WikiMultiHopQA dataset.

Model	MusiQue			
	200*7	200*12	500*3	500*5
# RAG Base (Vanilla RAG) #				
ChatGLM3-6B-32k	25.51	25.91	24.31	25.63
Qwen1.5-7B-32k	25.08	23.51	21.08	22.05
Vicuna-v1.5-7B-16k	15.68	14.55	16.05	13.89
Llama3-8B-8k	19.66	23.65	19.33	22.51
GPT-3.5-Turbo	25.22	28.23	25.34	27.06
GPT-3.5-Turbo-16k	21.84	25.41	24.80	23.79
Llama3-70B-8k	25.49	27.72	23.05	24.13
GLM-4	27.55	33.93	27.92	27.56
# Ours with SFT #				
LongRAG-ChatGLM3-6B-32k	33.00	33.12	30.09	31.98
LongRAG-Qwen1.5-7B-32k	31.85	32.22	27.25	25.84
LongRAG-Vicuna-v1.5-7B-16k	28.29	33.76	29.42	29.89
LongRAG-Llama3-8B-8k	31.70	38.19	33.90	29.57
# Ours without SFT #				
LongRAG-GPT-3.5-Turbo	32.83	32.64	29.83	28.03
LongRAG-GPT-3.5-Turbo-16k	30.37	32.11	28.96	26.58
LongRAG-GLM-4	38.40	39.68	34.67	33.05

Table 8: Overall performance of our LongRAG on MusiQue dataset.

Generator	HotpotQA		
	R&B	E&F w/o SFT	E&F w/ SFT (ChatGLM3-6b-32k)
LongRAG-GPT-3.5-Turbo-16k	50.17	59.11	57.82
LongRAG-GPT-3.5-Turbo	52.31	56.17	59.09
LongRAG-GLM-4	57.41	62.11	59.20

Table 9: Analysis of the component transferability of E&F on HotpotQA dataset.

Generator	2WikiMultiHopQA		
	R&B	E&F w/o SFT	E&F w/ SFT (ChatGLM3-6b-32k)
LongRAG-GPT-3.5-Turbo-16k	45.32	51.25	57.86
LongRAG-GPT-3.5-Turbo	43.44	51.37	54.62
LongRAG-GLM-4	52.91	57.16	55.96

Table 10: Analysis of the component transferability of E&F on 2WikiMultiHopQA dataset.

Generator	MusiQue		
	R&B	E&F w/o SFT	E&F w/ SFT (ChatGLM3-6b-32k)
LongRAG-GPT-3.5-Turbo-16k	21.84	30.37	34.52
LongRAG-GPT-3.5-Turbo	25.22	32.83	34.28
LongRAG-GLM-4	27.55	38.40	36.89

Table 11: Analysis of the component transferability of E&F on MusiQue dataset.

Model	HotpotQA				
	R&B	R&L	Ext.	Fil.	E&F
LongRAG-ChatGLM3-6B-32k w/ SFT	2181	10669	2254	1160	1233
LongRAG-Qwen1.5-7B-32k w/ SFT	2181	10669	2248	1260	1327
LongRAG-Vicuna-v1.5-7B-16k w/ SFT	2181	10596	2270	1233	1321
LongRAG-Llama3-8B-8k w/ SFT	2181	7428	2243	1101	1163

Table 12: Values of the token length fed into the generator on HotpotQA dataset.

Model	2WikiMultiHopQA				
	R&B	R&L	Ext.	Fil.	E&F
LongRAG-ChatGLM3-6B-32k w/ SFT	2086	8096	2171	937	1022
LongRAG-Qwen1.5-7B-32k w/ SFT	2086	8096	2162	941	1016
LongRAG-Vicuna-v1.5-7B-16k w/ SFT	2086	8096	2176	937	1027
LongRAG-Llama3-8B-8k w/ SFT	2086	6744	2150	813	876

Table 13: Values of the token length fed into the generator on 2WikiMultiHopQA dataset.

Model	MusiQue				
	R&B	R&L	Ext.	Fil.	E&F
LongRAG-ChatGLM3-6B-32k w/ SFT	2141	15062	2217	975	1051
LongRAG-Qwen1.5-7B-32k w/ SFT	2141	15062	2198	1050	1108
LongRAG-Vicuna-v1.5-7B-16k w/ SFT	2141	14520	2240	995	1094
LongRAG-Llama3-8B-8k w/ SFT	2141	7711	2196	828	883

Table 14: Values of the token length fed into the generator on MusiQue dataset.

Datasets	HotpotQA	2WikiMultiHopQA	MusiQue	QASPER
Num of long-context extractor data	200	200	200	100
Num of CoT-guiding data	200	200	200	100
Num of filtering data	200	200	200	-
Num of task-oriented data	200	200	200	-
Num of samples	800	800	800	200

Table 15: Statistics of our fine-tuning instruction dataset LRGinstruction.

[STEP-1]: Data construction prompt for Extractor
<p>{supporting paragraphs}</p> <p>Based on the above background only, please output the original information that needs to be cited to answer the following questions. Please ensure that the information cited is detailed and comprehensive.</p> <p>Question:{question}</p> <p>Output only the original information of the required reference: {global information}</p>
[STEP-2]: An LLM-based self-evaluator for Extractor
<p>I am going to provide you with a question, the background information, and the answer to that question. Please evaluate whether the answer can be solely derived from the given background information. If it can, set the status value as True, if it can't, set the status value as False.</p> <p>Question:{question}</p> <p>Background Information:{global information}</p> <p>Answer:{answer}</p> <p>Your output format should be the following json format: status: {the value of status}</p>
[RESULT]: Long-Context Extractor Data for Extractor
<p>Instruction: {content}</p> <p>Based on the above background, please output the information you need to cite to answer the question below. {question}</p> <p>Output: {global information}</p>

Table 16: Data construction pipeline for extractor and format illustration of long-context extractor data.

[STEP-1]: Data construction prompt for CoT guidance stage
<p>{supporting paragraphs}</p> <p>Given question:{question}</p> <p>The answer is:{answer}</p> <p>Your task is to give your thought process for this given question based on the above information, only give me your thought process and do not output other information.</p> <p>Thought process: {CoT}</p>
[STEP-2]: An LLM-based self-evaluator for CoT guidance stage
<p>Question:{question}</p> <p>Thought process of the question:{CoT}</p> <p>Answer:{answer}</p> <p>Please evaluate whether the thought process of this question can explain the answer to this question. If it can explain the answer, set the value of status to True. If it cannot explain the answer, set the value of status to False. Your output format should be the following json format: status: {the value of status}</p>
[RESULT-1]: CoT-guiding Data for CoT guidance stage
<p>Instruction: {content}</p> <p>Please combine the above information and give your thought process for the following Question:{question}</p> <p>Output: {CoT}</p>
[RESULT-2]: Filtering Data for filtering stage
<p>Instruction: Given an article:{content} Question:{question} Thought process for the question:{CoT}</p> <p>Your task is to use the thought process provided to decide whether you need to cite the article to answer this question. If you need to cite the article, set the status value to True. If not, set the status value to False. Please output the response in the following json format: {"status": {the value of status}}</p> <p>Output: {status}</p>

Table 17: Data construction pipeline for filter, and format illustration of CoT-guiding and filtering data.

[RESULT]: Task-Oriented Data for RAG task

Instruction:

{content}

Based on the above information, Only give me the answer and do not output any other words.

Question:{question}

Output:

{answer}

Table 18: Data construction pipeline for RAG task, and format illustration of task-oriented data.

Instruction:

Alan Marshal (actor) Alan Marshal (29 January 1909 – 9 July 1961) was an actor who performed on stage in the United States and in Hollywood films. He was sometimes billed as Alan Marshall or Alan Willey. Hans Tambs Lyche Hans Tambs Lyche (21 November 1859 – 16 April 1898) was a Norwegian engineer, unitarian minister, journalist and magazine editor.

Alan Deyermond Alan Deyermond FBA (24 February 1932 – 19 September 2009) was a British professor of Medieval Spanish Literature and Hispanist. His obituary cited him as " the English- speaking world's leading scholar of medieval Hispanic literature. " He spent his academic career associated with one University of London college, Westfield College (later merged with Queen Mary College, to form Queen Mary and Westfield College).

Deyermond started his career in 1955 as a lecturer at Westfield College, London. When Westfield merged with Queen Mary College in 1992, he moved to the Mile End site. In the period 1978 – 1980 he held a joint chair at Westfield and at Princeton University.

Ralph Tambs-Lyche Ralph Tambs- Lyche (6 September 1890 – 15 January 1991) was a Norwegian mathematician. He was born in Macon, Georgia as a son of Norwegian father Hans Tambs Lyche (1859 – 1898) and American mother Mary Rebecca Godden (1856 – 1938). He moved to Norway at the age of two.

.....

Alan M. Wald Alan Maynard Wald, usually Alan M. Wald or Alan Wald, is an American professor emeritus of English Literature and American Culture at the University of Michigan, Ann Arbor, and writer of 20th-Century American literature who focuses on Communist writers; he is an expert on the American 20th-Century "Literary Left. " Allen Sangree Allen Luther Sangree, also as Allan or Alan (c. 1878 – March 2, 1924) was an American sports writer and war journalist.

Based on the above background, please output the information you need to cite to answer the question below.

Who died earlier, Hans Tambs Lyche or Alan Deyermond?

Output:

Hans Tambs Lyche (21 November 1859 – 16 April 1898) was a Norwegian engineer, unitarian minister, journalist, and magazine editor. Alan Deyermond (24 February 1932 – 19 September 2009) was a British professor of Medieval Spanish Literature and Hispanist.

Table 19: An example of long-context extractor data of LRGInstruction.

Instruction:

Beatrice of Castile, Marchioness of Montferrat
Beatrice of Castile (5 November or 6 December 1254 – 1286) was a daughter of Alfonso X of Castile and his wife Violant of Aragon. She was Marchioness of Montferrat by her marriage and was mother of Irene of Montferrat.

Theodred II (Bishop of Elmham)
Theodred II was a medieval Bishop of Elmham. The date of Theodred's consecration unknown, but the date of his death was sometime between 995 and 997. Irene of Montferrat
Yolande of Montferrat (– 1317 in Constantinople) (also known as Violante, then Empress Irene) was the second wife of Andronikos II Palaiologos and thus empress of the Byzantine Empire. She was the heir of the Margraviate of Montferrat. Born in Casale, she was daughter of William VII, Marquess of Montferrat and his second wife Beatrice of Castile. Her maternal grandparents were King Alfonso X of Castile and his wife Violante of Aragon. Yolande (variation of Violante) was named after her grandmother.

In 1284, Andronikos II, a widower by his first marriage with Anna of Hungary, married Yolanda (who was renamed Eirene as Empress). She and Andronikos II were distant cousins, both being descendants of Andronikos Doukas Angelos (ca. 1122–1185). With her, Eirene brought the Montferrat rights to the kingdom of Thessalonica, a dominion that, despite having been conquered half-a-century before Eirene's birth by the Byzantine state of Epirus, was still claimed by its short-lived (1204–1224) Montferrat royal dynasty.

.....

Henry, Marquess of Montferrat
Henry(c.1020 –c.1044/5) was a member of the Aleramid dynasty. He was a younger son of William III of Montferrat and Waza. From 1042, he was co-ruler of the March of Montferrat with his older brother Otto II of Montferrat. Probably in 1041, certainly before 19 January 1042, he married Adelaide of Susa, the heiress of the March of Turin, which temporarily united the two great northwestern Italian marches of Turin and Montferrat.

Please combine the above information and give your thought process for the following

Question: What is the date of death of Irene Of Montferrat's father?

Output:

To find the date of death of Irene Of Montferrat's father, we can look at the information provided about William VII, Marquess of Montferrat. It states that he died on 6 February 1292. Since Irene was born around 1253, William VII would have been her father. Therefore, the date of death of Irene Of Montferrat's father is 6 February 1292.

Table 20: An example of CoT-guiding data of LRGinstruction.

Instruction:

Given an article:"Awaiting on You All" is a song by English musician George Harrison, released on his 1970 triple album, "All Things Must Pass". Along with the single "My Sweet Lord", it is among the more overtly religious compositions on "All Things Must Pass", and the recording typifies co-producer Phil Spector's influence on the album, due to his liberal use of reverberation and other Wall of Sound production techniques.

Harrison recorded the track in London backed by musicians such as Eric Clapton, Bobby Whitlock, Klaus Voormann, Jim Gordon and Jim Price – many of whom he had toured with, as Delaney & Bonnie and Friends, in December 1969, while still officially a member of the Beatles. Musically, the composition reflects Harrison's embracing of the gospel music genre, following his production of fellow Apple Records artists Billy Preston and Doris Troy.

.....

A similarly well-regarded live version, with backing from a large band including Clapton, Ringo Starr, Preston and Jim Keltner, was released on the 1971 album "The Concert for Bangladesh" and appeared in the 1972 film of the same name. Harrison's posthumous compilation (2012) includes a demo version of the song, recorded early in the 1970 sessions for "All Things Must Pass".

Question: What is the date of death of the performer of song Awaiting On You All?

Thought process for the question:The question asks for the date of death of the performer of the song "Awaiting on You All." We know from the given information that the song was written and performed by English musician George Harrison. To find his date of death, we can look for the date of death of George Harrison in the text. We find that George Harrison died on 29 November 2001. Therefore, the answer to the question is 29 November 2001.

Your task is to use the thought process provided to decide whether you need to cite the article to answer this question. If you need to cite the article, set the status value to True. If not, set the status value to False. Please output the response in the following json format:
{"status": {the value of status}}

Output:

```
{"status": {"True"}}
```

Table 21: An example of filtering data of LRGInstruction.

Instruction:

My Name Is Anthony Gonsalves (film) My Name Is Anthony Gonsalves is a Bollywood drama film starring newcomer Nikhil Dwivedi, Amrita Rao and Mithun Chakraborty as the lead protagonists. The film is directed by Eeshwar Nivas. The name of the movie is derived from the 1977 hit movie Amar Akbar Anthony's famous song, " My Name Is Anthony Gonsalves." It was released on 11 January 2008 and was a box office bomb.

My Name Is JuaniMy Name Is Juani is a 2006 Spanish drama film written and directed by Bigas Luna. My Name Is BanduMy Name is Bandu is a 2015 Sri Lankan Sinhala comedy, family film directed by Suranga de Alwis and produced by Suranga de Alwis. It stars Bandu Samarasinghe, and Anusha Damayanthi in lead roles along with Rodney Warnakula, Roy de Silva and Mark Samson. Music for the film is done by Sarath de Alwis. The film is the 85th film of Bandu Samarasinghe. It is the 1239th Sri Lankan film in the Sinhala cinema.

My Name Is KhanMy Name Is Khan is a 2010 Indian Hindi- language drama film directed by Karan Johar, produced by Hiroo Johar and Gauri Khan, and starring Shah Rukh Khan and Kajol in lead roles.

.....

The film stars Shakib Khan and Sahara in the lead roles, with Ahmed Sharif, Misha Shoudagor, Probir Mitro and Rahena Joli playing other significant roles in the film.

My Name Is Sultan was released on 20 August 2012. Leslie, My Name Is EvilLeslie, My Name Is Evil is a 2009 Canadian film written and directed by Reginald Harkema. It was renamed" Manson, My Name Is Evil" after its initial release.

My Name Is NobodyMy Name Is Nobody is a 1973 comedy spaghetti western starring Terence Hill and Henry Fonda. The film was directed by Tonino Valerii.

My Name Is Rocco PapaleoMy Name Is Rocco Papaleo is a 1971 Italian comedy film directed by Ettore Scola.

Based on the above information, Only give me the answer and do not output any other words.

Question: Which film was released more recently, My Name Is Bandu or Leadbelly (Film)?

Answer:

Output:

My Name Is Bandu

Table 22: An example of task-oriented data of LRGinstruction.

Prompt of LLM-augmented information extractor
<p>Instruction: {content} Based on the above background, please output the information you need to cite to answer the question below. {question}</p> <p>Output: {global information}</p>
Prompt of CoT guidance stage in CoT-guided filter
<p>Instruction: {content} Please combine the above information and give your thought process for the following Question:{question}</p> <p>Output: {CoT}</p>
Prompt of filtering stage in CoT-guided filter
<p>Instruction: Given an article:{content} Question:{question} Thought process for the question:{CoT}</p> <p>Your task is to use the thought process provided to decide whether you need to cite the article to answer this question. If you need to cite the article, set the status value to True. If not, set the status value to False. Please output the response in the following json format: {"status": {the value of status}}</p> <p>Output: {status}</p>
Prompt of LLM-augmented generator
<p>Instruction: {content} Based on the above information, Only give me the answer and do not output any other words. Question:{question}</p> <p>Output: {answer}</p>

Table 23: All prompts of LongRAG system.

Question: Where did the performer of song I'll Say It graduate from?

Input to generator (2082 tokens):

Answer the question based on the given passages. Only give me the answer and do not output any other words. The following are given passages. The duo promoted the song by performing it on various television shows and at various venues, of which included GMTV and Sony Ericsson's Dance Nation Festival. This was planned to be the first single off the band's second studio album Say It Now, which was scheduled for release in November 2009, but due to the low chart placing of "Say It", the album was eventually cancelled. Background "Say It" was written by Carl Björnell, Didrik Thott and Sebastian Thott.

.....

We just want to show progression."The song was composed in a key of C sharp minor and runs at a tempo of 126.96 beats per minute. The song was produced with consistence of various drum and bass and electronica instrumentation. Passage 1: **I'll Say It** "I'll Say It" is a **song** written by American musician Adam Schlesinger and **recorded by** comedian Kathy **Griffin**, released as the theme song for her show, **Kathy**. It was additionally used as the introduction music to her 2012 comedy special "Kennedie Center on Hers" and continued to be used in future specials. **On August 20, 2012, Griffin released a seven track EP containing dance remixes of "I'll Say It"**. Music video The music video begins in the day with Kathy Griffin in her house preparing her make-up. It shows her daily routine visiting her dogs, leaving the house and driving to a theater, ending with her on stage in her signature pose. The scenes are interlaced with various clips of Los Angeles, California. Passage 10: Say It (Booty Luv song) "Say It" is a song by female English dance music duo **Booty Luv**.

.....

Filmography Film Television Other Stand-up specials Discography On June 10, 2008, Griffin released a comedy CD titled For Your Consideration. The disc was recorded at the ETK Theatre at the Grand Theatre Center For The Arts in Tracy, California on February 17, 2008. Griffin stated she decided to release the CD to try to win a Grammy award. On August 25, 2009, Griffin released a second comedy album, Suckin' It for the Holidays, in another bid for a Grammy. **Griffin received her third Grammy nomination for Kathy Griffin: Does the Bible Belt in 2010**. **On May 4, 2012, the full length version of "I'll Say It", the theme song of her show Kathy, was released to iTunes as a single. On August 20, 2012, Griffin released a seven-track EP containing dance remixes of "I'll Say It"**. Bibliography Official Book Club Selection: A Memoir According to Kathy Griffin. Ballantine Books. 2009. ISBN 978-0345518569. Kathy Griffin's Celebrity Run-Ins: My A-Z Index. Flatiron Books. 2016. ISBN 978-1250115638. Song went on a five-year hiatus from acting. She became an adjunct professor and part-time lecturer at **Seoul Arts College** in 2010, as a faculty member of the Department of Performing Arts and the Department of Broadcasting, Entertainment and Visual Arts.

.....

Asher Roth sampled the song for his debut rap single "I Love College". After the song leaked onto the internet, Rivers Cuomo reportedly refused to clear the sample, which prompted Roth to debut a remixed version of his song as his official debut single. Answer the question based on the given passages. Only give me the answer and do not output any other words.

Question: Where did the performer of song I'll Say It graduate from?

Answer:

Answer of RAG-base: **Seoul Arts College** ✗

Golden Answer: **Lee Strasberg Theatre and Film Institute** ✓

Wrong Reason: **Incomplete key information**

Table 24: A question-answering example of Vanilla RAG (RAG-Base). **The words in the green area** indicate correct relevant information and answers while **red** means the opposite. **The blue snippets** are question-relevant information. The correct answer is labeled "✓", while wrong answer labeled "✗".

Question: Where did the performer of song I'll Say It graduate from?

Input to generator (23047 tokens):

Answer the question based on the given passages. Only give me the answer and do not output any other words. The following are given passages.

.....

The girls then head downstairs to a mini casino where they gamble. The girls are then seen against various backgrounds and laying on chairs. Finally, the girls have a party in their hotel room and invite their friends and some men to their hotel rooms, before sending them away. Chart performance Weekly charts Year-end charts Passage 1: **I'll Say It** "I'll Say It" is a **song** written by American musician Adam Schlesinger and **recorded by** comedian Kathy **Griffin**, released as the theme song for her show, *Kathy*. It was additionally used as the introduction music to her 2012 comedy special "Kennedie Center on Hers" and continued to be used in future specials. **On August 20, 2012, Griffin released a seven track EP containing dance remixes of "I'll Say It"**. Music video The music video begins in the day with Kathy Griffin in her house preparing her make-up. It shows her daily routine visiting her dogs, leaving the house and driving to a theater, ending with her on stage in her signature pose. The scenes are interlaced with various clips of Los Angeles, California. Charts Passage 2: Kathy Griffin Kathleen Mary Griffin (born November 4, 1960) is an American comedian and actress who has starred in television comedy specials and has released comedy albums. In 2007 and 2008, Griffin won Primetime Emmy Awards for her reality show *Kathy Griffin: My Life on the D-List*. She has also appeared in supporting roles in films. **Griffin was born in Oak Park, Illinois. In 1978, she moved to Los Angeles, where she studied drama at the Lee Strasberg Theatre and Film Institute and became a member of the improvisational comedy troupe The Groundlings.** In the 1990s, Griffin began performing as a stand-up comedian and appeared as a guest star on television shows, including a supporting role on the NBC sitcom *Suddenly Susan* (1996–2000).

.....

Griffin released a second comedy album, *Suckin' It for the Holidays*, in another bid for a Grammy. Griffin received her third Grammy nomination for *Kathy Griffin: Does the Bible Belt* in 2010. On May 4, 2012, the full length version of "I'll Say It", the theme song of her show *Kathy*, was released to iTunes as a single. On August 20, 2012, Griffin released a seven-track EP containing dance remixes of "I'll Say It".

.....

Song Yoon-ah was born in Seoul, but spent her childhood in Gimcheon, North Gyeongsang Province. She has two elder brothers, the first one is a doctor. While studying Cultural Anthropology as a freshman at **Hanyang University**, she was recommended by an older schoolmate to a modeling agency.

.....

Chiptune artist Inverse Phase parodied the song on a Commodore 64, titling it "Say It Ain't Sixty-FO" Calpurnia covered the song for Spotify's Under Cover podcast in 2018 In popular culture "Say It Ain't So" is a playable track in the video games *Rock Band* and *Rocksmith 2014* in addition to appearing on an episode of *Hindsight*. Answer the question based on the given passages. Only give me the answer and do not output any other words.

Question: Where did the performer of song I'll Say It graduate from?

Answer:

Answer of RAG-Long: **Hanyang University** ✗

Golden Answer: **Lee Strasberg Theatre and Film Institute** ✓

Wrong Reason: **Complete key information but lost in middle**

Table 25: A question-answering example of our LongRAG with RAG-Long component strategy. **The words in the green area** indicate correct relevant information and answers while **red** means the opposite. **The blue snippets** are question-relevant information. The correct answer is labeled "✓", while wrong answer labeled "✗".

Question: Where did the performer of song I'll Say It graduate from?

Input to generator (644 tokens):

Answer the question based on the given passages. Only give me the answer and do not output any other words. The following are given passages. Passage 1: I'll Say It "I'll Say It" is a song written by American musician Adam Schlesinger and recorded by comedian Kathy Griffin, released as the theme song for her show, Kathy. It was additionally used as the introduction music to her 2012 comedy special "Kennedie Center on Hers" and continued to be used in future specials. On August 20, 2012, Griffin released a seven track EP containing dance remixes of "I'll Say It". Music video The music video begins in the day with Kathy Griffin in her house preparing her make-up. It shows her daily routine visiting her dogs, leaving the house and driving to a theater, ending with her on stage in her signature pose. The scenes are interlaced with various clips of Los Angeles, California. in a ceremony officiated by comedian Lily Tomlin. Filmography Film Television Other Stand-up specials Discography On June 10, 2008, Griffin released a comedy CD titled For Your Consideration. The disc was recorded at the ETK Theatre at the Grand Theatre Center For The Arts in Tracy, California on February 17, 2008. Griffin stated she decided to release the CD to try to win a Grammy award. On August 25, 2009, Griffin released a second comedy album, Suckin' It for the Holidays, in another bid for a Grammy. Griffin received her third Grammy nomination for Kathy Griffin: Does the Bible Belt in 2010. On May 4, 2012, the full length version of "I'll Say It", the theme song of her show Kathy, was released to iTunes as a single. On August 20, 2012, Griffin released a seven-track EP containing dance remixes of "I'll Say It". Bibliography Official Book Club Selection: A Memoir According to Kathy Griffin. Ballantine Books. 2009. ISBN 978-0345518569. Kathy Griffin's Celebrity Run-Ins: My A-Z Index. Flatiron Books. 2016. ISBN 978-1250115638. The performer of the song "I'll Say It" is Kathy Griffin, an American comedian and actress who has starred in television comedy specials and has released comedy albums. She attended the Lee Strasberg Theatre and Film Institute in Los Angeles, where she studied drama. Answer the question based on the given passages. Only give me the answer and do not output any other words.

Question: Where did the performer of song I'll Say It graduate from?

Answer:

Answer of LongRAG: Lee Strasberg Theatre and Film Institute ✓

Golden Answer: Lee Strasberg Theatre and Film Institute ✓

Table 26: A question-answering example of our LongRAG system with E&F component strategy. The words in the green area indicate correct relevant information and answers while red means the opposite. The blue snippets are question-relevant information. The correct answer is labeled "✓", while wrong answer labeled "✗".