Long Context vs. RAG for LLMs: An Evaluation and Revisits

Anonymous ACL submission

Abstract

Extending context windows (*i.e.,* Long Context, LC) and using retrievers to selectively access relevant information (*i.e.,* Retrieval-Augmented Generation, RAG) are the two main strategies to enable LLMs to incorporate extremely long external contexts. This paper revisits recent studies on this topic, highlighting their key insights and discrepancies. We then provide a more comprehensive evaluation by filtering out questions answerable without external context, identifying the most effective retrieval methods, and expanding the datasets. We show that LC generally outperforms RAG in question-answering benchmarks, especially for Wikipedia-based questions. Summarization-based retrieval performs comparably to LC, while chunk-based retrieval lags behind. However, RAG has advantages in dialogue-based and general question queries. These insights underscore the trade-offs between RAG and LC strategies, offering guidance for future optimization of LLMs with external knowledge sources. We also provide an in-depth discussion on this topic, highlighting the overlooked importance of context relevance in existing studies.

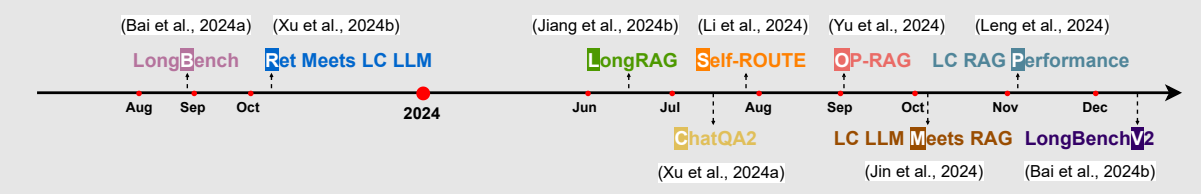
1 Introduction

Large Language Models (LLMs) (Brown et al., 2020) have demonstrated strong zero/few-shot capabilities in open-ended question answering (Yang et al., 2019). However, they face challenges such as hallucinations (Shuster et al., 2021; Ji et al., 2023), lacking real-time information and domain-specific knowledge (Su et al., 2024; Zhang et al., 2024), among others. A common solution is to enhance LLMs with external memory to provide reliable and up-to-date data sources. Yet, incorporating additional content is constrained by the limited context window of LLMs. To address this, two main approaches are adopted: (i) building models with long context windows to read in more information (**LC**) (Fei et al., 2024; Chen et al., 2023; Wang et al., 2024c), and (ii) employing retrievers to include text segments relevant to the query (**RAG**) (Jiang et al., 2023; Asai et al., 2024; Gao et al., 2023).

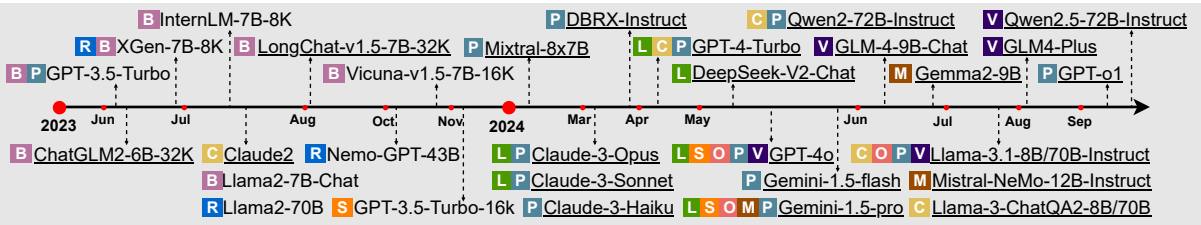
As shown by the timeline in Figure 1a, there is a clear trend toward developing models that handle longer context windows and combining LC with RAG methods. The chronological overview of related studies highlights an increasing focus on both LC and RAG since mid-2023, as evidenced by a growing number of publications aimed at optimizing the efficient retrieval, and utilization of long contexts. The development of models supporting longer context windows underscores the growing importance of handling extensive inputs effectively.

Despite the broad consensus regarding the importance of LC and RAG, there remain disagreements and contradictory insights from different studies, summarized in Table 1. For example, while several studies agree on the effectiveness of combining LC and RAG (Xu et al., 2024b; Jiang et al., 2024b), others suggest that combining may not be beneficial (Bai et al., 2024a; Jin et al., 2024). Moreover, conflicting conclusions are reported regarding the benefits of RAG versus LC. Some papers find RAG advantageous in certain contexts (Xu et al., 2024a; Yu et al., 2024), while others highlight superior results from LC (Li et al., 2024; Xu et al., 2024b). These divergent insights showcase the complexity and ongoing debates in the field, suggesting that optimal strategies may vary depending on specific model architectures and benchmark conditions.

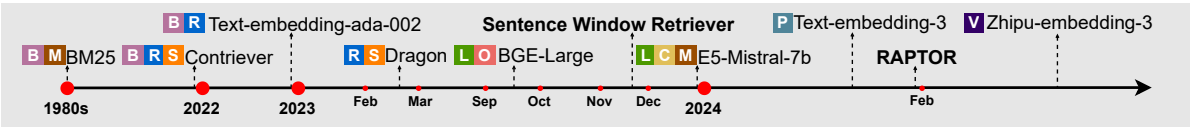
To explore the underlying reasons, we conduct an in-depth investigation into the conditions that lead to disagreements among existing studies. During this process, we also identify key aspects that may have been overlooked in earlier research. Specifically, we revisit the evaluation process and implement the following changes. To build our



(a) Related work on LC and RAG, each paper is labeled by a char and one color. For instance, green and "L" represent "LongRAG".



(b) Chronological progress of key LLMs from 2023 to 2024. We focus on the models that publications in 1a use. We underline the models that support context window length of $\geq 32K$.



(c) History of frequently used retrievers from the 1980s until 2024. We **bold** the retrievers that no existing publications in 1a uses.

Figure 1: Chronological overview of the development of RAG and LC. The Sub-graphs respectively illustrate the timelines for (a) publications related to LC and RAG, (b) long-context models, and (c) retrievers. We label before each model and retriever with the char and color block representing the publication that uses it.

benchmark, we (i) remove context-independent questions to focus on externally grounded QA, (ii) evaluate retrievers on a core 1,000+ question subset to select the strongest baseline, and (iii) expand the filtered set tenfold by collecting all original passages from the 12 QA datasets[1]. This unified dataset supports all downstream LC vs. RAG comparisons and in-depth analysis, providing better statistical significance and reducing bias.

Our key contributions in this paper are: (i) Providing a comprehensive survey of existing studies on LC and RAG, analyzing their implementations and key insights (appendix A). (ii) Constructing a large-scale, filtered QA dataset and selecting the strongest retriever on a core 1,000+ question subset (§ 2) (iii) Proposing a fair evaluation framework and delivering in-depth comparisons of LC and RAG across multiple settings (§ 3) (iv) Discussing challenges for comparing and combining LC and RAG, reflecting on the key points that researchers tend to overlook in this field.(§ 5 and appendix D) Evaluation results indicate that LC models generally outperform RAG when processing self-contained information like stories, while RAG excels at handling fragmented information, particularly in dialogue-based contexts. These experiments deepen our understanding of the strengths and limitations of LC and RAG, offering valuable insights into optimizing retrieval strategies and effectively integrating these approaches to enhance performance in open-domain question answering.

2 Question Filtering and Expansion

To ensure a fair and comprehensive comparison, we curate our evaluation dataset based on existing datasets, and apply necessary filtering (§ 2.1) and augmentation (§ 2.2). We select 12 long-context QA datasets frequently used in studies comparing LC and RAG: Natural Questions (Kwiatkowski et al., 2019), 2WikiMultihopQA (Ho et al., 2020), HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), MultiFieldQA (Bai et al., 2024a), NarrativeQA (Kočiský et al., 2018), QASPER (Dasigi et al., 2021), QuALTY (Pang et al., 2022), Coursera, TOEFL-QA, and MultiDoc2Dial (An et al., 2024). We also include the NovelQA (Wang et al.,

[1]The experiment code and expanded datasets will be released upon acceptance.

Paper	Type	Findings
LongBench (B) (Bai et al., 2024a)	•	Retrieval helps 4k model, but not 16k/32k models.
	+	Models benefit from continuous training on long contexts.
	+	Splitting context into shorter and more chunks is better.
Ret-LC LLM (R) (Xu et al., 2024b)	⋆	LC is better for multi-hop benchmarks than 4k RAG.
	∘	RAG improves on 70B/43B models on all context lengths.
	+	For LC model, best results are obtained from top-5 or top-10.
LongRAG (L) (Jiang et al., 2024b)	∘	Retrieval benefits from long retrieval units.
ChatQA2 (C) (Xu et al., 2024a)	☆	For sequence lengths up to 32K, RAG outperforms LC.
	∘	From 3K to 24K, greater context window benefits RAG.
Self-ROUTE (S) (Li et al., 2024)	⋆	LC consistently outperforms RAG, but RAG has lower cost.
OP-RAG (O) (Yu et al., 2024)	☆	Efficient retrieval can outperform brute-force LC.
	+	Too many chunks in RAG harms performance.
	+	Preserving the original order is better than ordering by score.
LC LLM-RAG (M) (Jin et al., 2024)	•	Retrieve more passages first improves performance then drops.
	+	Ordering higher score information to front and back helps.
LC RAG Performance (P) (Leng et al., 2024)	∘	Most close models' RAG improves up to 100k tokens.
	•	Most open models' RAG peak at 16k-32k then performance drops.
LongBench v2 (V) (Bai et al., 2024b)	☆	GPT-4o performs better at 128k without RAG.
	∘	GPT-4o performance keeps increasing to 128k RAG context.
	•	Qwen2.5 & GLM-4-Plus drop with >32k RAG contexts.

Table 1: Important findings from existing studies that compare or combine LC with RAG (label in brackets). We group the insights into three categories: 1) General strategies that improve performance marked by +. 2) Combining LC and RAG, where ∘ indicates combining is good, and • for combining is not helpful, and 3) Comparing LC and RAG, where ☆ indicates RAG outperforms LC, and ⋆ for LC outperforms RAG.

2024a) dataset, a high-quality, human-annotated resource derived from long-form novels. We present an overview of these datasets in Table 2, including their type, context type (single-doc or multi-doc), context source, average context length, and representative studies that have utilized each dataset.

2.1 Question Filtering

Given the strong capabilities of modern LLMs, many questions can be directly answered based on knowledge encoded in their parameters (Basmova et al., 2024), reducing the need for external context in some cases. However, certain queries, such as those related to private conversations, will always require additional context. To determine which approach more effectively enhances an LLM's performance with long documents, we filter the datasets to include only questions that the LLM cannot answer correctly without external context. This ensures that any correct answers obtained subsequently must rely on external knowledge rather than the model's built-in knowledge.

For our implementation, we use GPT-4o for question filtering due to its strong capabilities. We employ a strict exact-match scoring metric to ensure that the model not only provides the correct answer but also demonstrates a complete understanding of the required information.

2.2 Question (and Context) Expansion

RAG and LC produce identical answers for about 60% of the questions in existing evaluations (Li et al., 2024), leaving relatively few questions to help us understand the differences between the two. To ensure robust statistical significance, we expand the dataset size to approximately 20,000 questions by collecting additional samples.

To maintain a similar distribution as the original datasets, we follow two principles during data collection. First, we collect questions only from the original source of each dataset, avoiding artificially generated or LLM-augmented questions. Second, we add distracting passages to the original context for each question to extend the context length, following the implementation described in LongBench. For NovelQA, we use all its available questions. For Coursera, MultiFieldQA, and MultiDoc2Dial datasets, we do not further enlarge their sizes to avoid introducing artificial data.

Hereafter, we refer to the expanded dataset as the

Dataset	T	Doc	Source	Avg Len	Used by Papers	# Q	# Kept	% Kept	Mode
NQ	K	multi	Wikipedia	18,164.7	M, P	109	22	20	Open
Coursera	K	multi	Coursera	7,934.3	NIL (L-eval)	172	54	32	MCQ
NovelQA	C	single	books	67,000.0	NIL (NovelQA)	210	109	52	MCQ
2WikiMHQA	R	multi	Wikipedia	7,191.3	B, S, M	300	152	51	Open
HotpotQA	R	multi	Wikipedia	10,602.7	B, R, L, C, S, M	200	93	47	Open
MuSiQue	R	multi	Wikipedia	12,974.3	B, R, C, S	200	140	70	Open
MultiFieldQA	C	single	papers, reports	5,706.1	B, R, L, C, S	150	121	81	Open
NarrativeQA	C	single	books, films	25,274.2	B, R, S	200	171	86	Open
QASPER	C	single	papers	5,350.3	B, R, C	224	221	99	Open
QuALTY	C	single	stories	5,089.2	R, C	202	202	100	MCQ
TOEFL-QA	C	single	exams	729.1	NIL (L-eval)	121	121	100	MCQ
MultiDoc2Dial	C	multi	dialogue	3,076.9	NIL (L-eval)	158	158	100	Open

Table 2: Overview of the original datasets (i.e., the pre-expanded *sample question set*) and their characteristics. The column "T" represents dataset type with values "K" for "Knowledge", "R" for "reasoning", and "C" for "reading comprehension". For each dataset, we report the existing papers (with the label) about LC & RAG that use it. If no paper has used it, we report its source like L-eval (An et al., 2024). We also report number of questions in each set (# Q), number and percentage of questions retained after filtering (# Kept and % Kept) out questions needing no context, and mode of question.

Dataset	# Questions	# Kept Q	% Kept Q
Coursera	172	54	32
NQ	1,109	373	34
NovelQA	2,283	869	38
2WikiMHQA	2,300	1,036	45
HotpotQA	2,200	1,113	51
MuSiQue	2,200	1,663	78
MultiFieldQA	150	121	81
NarrativeQA	2,211	1,880	85
QASPER	2,718	2,674	98
QuALTY	2,725	2,725	100
TOEFL-QA	962	962	100
MultiDoc2Dial	158	158	100
Total	**19,188**	**13,628**	**71**

Table 3: Statistics of the *full question set*, ordered by increasing percentage of questions kept after filtering out questions needing no context.

full question set and the original, pre-expansion dataset as the **sample question set**.

2.3 Dataset Statistics

After expansion, we obtain 19,188 questions, of which 13,651 require context to be answered using the filtering method from § 2.1, as listed in Table 3. Notably, questions grounded in factual knowledge, such as those from Coursera, show a high removal rate. Similarly, questions drawn from well-known books or requiring multi-hop reasoning often exhibit a higher likelihood of being directly answered by LLMs without context. Comparing the 12 individual datasets, we observe a similar filtering rate between the sample and the full question sets (see Tables 2 and 3), indicating that both sets follow a similar distribution.

3 Evaluation Methodology

3.1 Evaluation Framework

Our evaluation of RAG and LC is conducted in the following three phases.

Phase 1: Empirical Study on Retrievers. We evaluate five retrievers: BM25, Contriever, OpenAI Embeddings, Llama-Index, and RAPTOR, on the sample question set. The retriever yielding the best performance is then selected for subsequent comparisons with LC on the full question set.

Phase 2: Comparing RAG and LC. Using the best retriever, RAG is compared with LC by answering questions on the full question set. Both methods use the same underlying LLM for question answering. For RAG, relevant documents or chunks are fetched from the available context and provided to the LLM as input to generate answers. In contrast, for LC, the entire context available to the question is given to the LLM, with truncation from the back of the context applied if the context exceeds the model's context window. The evaluation metrics are explained in § 3.3.

Phase 3: In-depth Analysis. We focus on 4 specific subsets of questions: 1) those answered correctly only by RAG, 2) those answered correctly only by LC, 3) those RAG gives better answers, and 4) those LC gives better answers. These subsets are analyzed to understand the types of questions each method excels at, providing insights into the strengths and limitations of both approaches in different scenarios.

3.2 Retriever Selection

Figure 1 shows that existing studies primarily select one or more chunk-based retrieval methods, while index- and summarization-based retrievers are less frequently evaluated. In our study, we evaluate various retrieval methods to ensure that RAG is supported by the most effective retrievers.

For chunk-based retrieval, we use BM25 (Robertson and Zaragoza, 2009), Contriever (Izacard et al., 2022), and OpenAI's text-embedding-3-Small. BM25 serves as a classic baseline, while Contriever and text-embedding-3-Small represent embeddings from well-performing closed-source and open-source models, respectively.

For index-based retrieval, we employ Llama-index and leverage two indexing methods that suit long documents. Specifically, **tree-index** organizes documents into a hierarchical tree structure, enabling efficient retrieval of context. The root node contains a high-level summary, while subsequent child nodes store progressively finer-grained representations. When queried, the retrieval process navigates through this hierarchy, starting from the top-level summary and moving down to more specific nodes as needed. **Sentence Window Retriever** focuses on local, sentence-level context rather than entire documents or large text chunks. It creates smaller "windows" of a few sentences each. When a query arrives, the retriever searches these windows to identify segments most semantically similar to the query. By working at a finer granularity, the sentence window retriever provides more targeted and contextually accurate snippets of text, improving the model's ability to answer specific questions.

For **summarization-based** retrieval, we use RAPTOR (Sarthi et al., 2024). It constructs a hierarchical tree by recursively clustering text chunks based on semantic similarity, summarizing each cluster into a parent node, and continuing this process until no further clustering is possible. After constructing the tree, we apply the collapsed tree traversal approach, as previous work has demonstrated its superior performance. This approach flattens the hierarchical structure into a single layer and compares the query against all nodes across every level simultaneously. The top-k most relevant nodes are then selected based on a predefined token limit, ensuring that the retrieved information maintains the appropriate level of granularity.

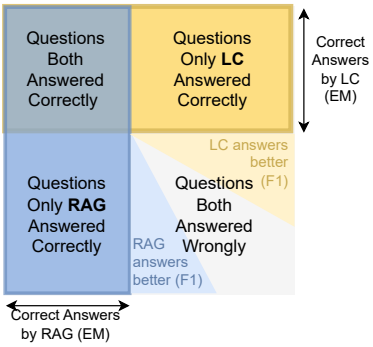


Figure 2: Evaluation Matrix for In-depth Analysis.

Although RAPTOR's implementation appears similar to the Llama Tree Index, they differ in both construction and navigation. First, Llama Tree Index groups consecutive nodes, while RAPTOR freely clusters nodes from far positions, and even allows a single node to appear in multiple clusters. Second, Llama Tree Index navigates down the hierarchy to retrieve only leaf nodes, while RAPTOR evaluates all nodes from all layers simultaneously. Hence, RAPTOR can retrieve not only original texts but also generated summaries.

3.3 Evaluation Metric

We use a win-lose rate system to compare LC and RAG, as illustrated in Figure 2. The horizontal yellow block represents the questions that the LLM answers correctly using LC, while the vertical blue block represents the questions that the LLM answers correctly using RAG. Their overlap in the top-left corner represents the questions that both methods answer correctly. We apply an *Exact Match* (EM) score strictly to all questions to determine the correctness of the answers. Excluding the overlap, the top right block indicates the questions that **only LC** answers correctly, and similarly, the bottom left block indicates the questions that **only RAG** answers correctly.

The remaining gray block represents the questions that both RAG and LC answer incorrectly, as judged by Exact Match. Since many questions involve long open-ended responses, we calculate the $F1$ scores of the answers provided by both methods against the ground truth. If RAG achieves a higher $F1$ score than LC, we consider RAG to have answered the question better, and vice versa for LC. A detailed explanation of $F1$ score calculation is provided in appendix B

The loose evaluation setting considers all cases

Dataset	# Questions	LC Correct	RAG Correct	LC Only	RAG Only	LC Better	RAG Better
Coursera	54	26	20	10	4	10	4
2WikiMHQA	1,036	594	431	242	79	265	107
HotpotQA	1,113	876	723	212	59	231	67
MultiFieldQA	121	63	60	14	11	44	21
NQ	373	189	138	75	24	104	35
NarrativeQA	1,880	558	405	276	123	685	281
QASPER	2,674	884	863	517	496	1,011	762
QuALITY	2,725	2,290	2,050	402	162	402	162
TOEFL-QA	962	895	884	26	15	26	15
MuiQue	1,663	821	663	344	186	426	225
MultiDoc2Dial	158	14	38	5	29	65	58
NovelQA	869	466	408	164	106	164	106
Overall	**13,628**	**7676**	**6,683**	**2,287**	**1,294**	**3,433**	**1,843**

Table 4: Performance of LC and RAG across different datasets. We report the number of questions answered correctly by each method, as well as the breakdown of questions where: only LC answers correctly (LC Only), only RAG answers correctly (RAG Only), LC outperforms RAG (LC Better), and RAG outperforms LC (RAG Better).

in which one method outperforms the other, including 1) when one method obtains the correct answer and the other is wrong under EM, and 2) when one method achieves a higher $F1$ score. We adopt this loose evaluation because references for some datasets are long, open-ended answers, making it very unlikely to match them exactly under EM. In addition, some short answers (about 5–6 words) may differ slightly from the reference while still conveying the correct idea. Although these answers would be marked incorrect by EM, they might attain a high $F1$ score. Hence, comparing $F1$ scores helps compensate for the strictness of EM.

4 Experiments

To obtain answers, we use the same prompt *"From the context: [context], answer the questions briefly with no explanation."* for both retrieval and long context settings. For MCQ questions, we add *"Answer the question with the letters of the correct options (e.g. A, BC, C, ACD, etc.) without including text"*. These prompts ensure LLMs to directly answer the questions, which makes evaluation more convenient. We use GPT-4o-2024-05-13 as evaluation backbone. Other than main experiment, we also perform additional analyses—including a word-frequency study that highlights LC's strength on narrative questions versus RAG's advantage on technical topics, an assessment of generator models confirming retrieval's dominant role, and case studies that show typical errors; **the full results are presented in appendix C**.

4.1 Phase 1: Retrievers

Evaluated on the sample question set, Table 5 reports the results of chunk-, index-, and summarization-based retrievers. Among them, RAPTOR performs the best with a correct answer rate of 38.5%, while Index-based retrievers outperform chunk-based retrievers. Within index-based retrievers, the "RAG Only" score for Tree Index is much lower than that for Window Parsing (82 vs. 91), and their "RAG Better" scores are nearly identical (234 vs. 237). This discrepancy suggests that Tree Index may be undervalued in the "RAG Only" metric but still contributes in open question scenarios that require long answers.

Type	Retriever	Correct (%)	RAG Only	RAG Better
Chunk	BM25	319 (20.4)	50	141
	Contriever	315 (20.1)	43	143
	Text-emb-3-small	338 (21.6)	47	151
Index	Tree Index	470 (30.1)	82	234
	Window Parsing	555 (35.5)	91	237
Summarization	RAPTOR	**602 (38.5)**	97	258

Table 5: Comparison of different retrieval methods

We further observe the questions and contexts that each retriever exclusively answers correctly. RAPTOR shows stronger ability than other retrievers, especially in scenarios that require an entire understanding of the document, like research papers. Chunk-based methods struggle when required information is spread across multiple chunks. Index-based retrievers are not as strong in overall understanding as RAPTOR, but they show good ability in interpreting dialogues. Therefore, we select RAPTOR as the primary retriever for evaluation on the full question set.

4.2 Phase 2: Comparing LC and RAG

We compare LC and RAG on the filtered, full question set. The results across 12 datasets are summarized in Table 4. Overall, LC correctly answers

56.3% of the questions, while RAG provides correct answers to 49.0%. LC correctly answers more than 2,000 questions that RAG misses, while RAG exclusively answers almost 1,300 questions. When looking at the loose evaluation setting, LC answers 3,433 questions better than RAG, and RAG answers 1,843 questions better than LC. The gap further widens compared to strict setting, indicating long-context LLM's ability to answer questions with open long answers is also strong.

Looking at individual datasets, in MultiDoc2Dial, RAG exhibits better performance than LC in strict evaluation (5 vs 29), but is surpassed by LC in loose evaluation (65 vs 58). In contrast, on datasets like NarrativeQA and QuaLITY, LC shows a strong lead not just in overall correctness but also in the number of questions that are answered better. Collectively, the results show that both methods have unique strengths and limitations.

Although LC shows better overall results than RAG, out of the 13,628 questions, almost 10% can be only answered correctly by RAG, which is not a small ratio. This shows that retrievers cannot be simply replaced by long-context LLM in searching. This also motivates us to further examine what kind of questions (and context) can be only answered correctly by RAG (or LC).

4.3 Phase 3: In-Depth Analysis

The overall results are influenced by the combined effects of different scenarios, so we need to separately analyze each scenario to see if more detailed results can be obtained. We analyze the performance of LC and RAG across different knowledge sources (Figure 3) and question types (Figures 4). Here, we use EM Scores only, for a strict evaluation standard. We also report the results for loose evaluation standard (i.e., EM Scores and $F1$ Scores) in appendix C.1, which shows similar trends.

From Figure 3, it is evident that LC excels with knowledge sources such as Wikipedia and stories. However, the Wikipedia context is collected by adding extensive noise to create long context, which generally makes the context less relevant to the question, with only a small portion being useful. This synthetic context formation partially simulates the RAG process and may introduce an unfair bias against the RAG pipeline. In addition, summarization-based retrieval methods may split Wikipedia articles unnaturally, generating less meaningful summaries. LC's strong performance demonstrates that long-context LLMs are robust to noise in such forms of context.

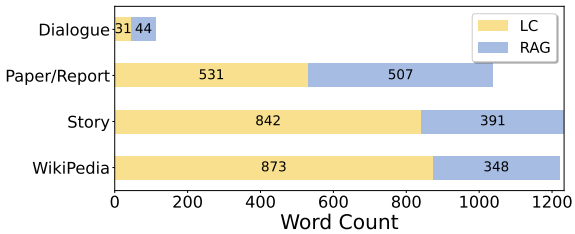


Figure 3: Performance breakdown by knowledge source for LC Only and RAG Only.

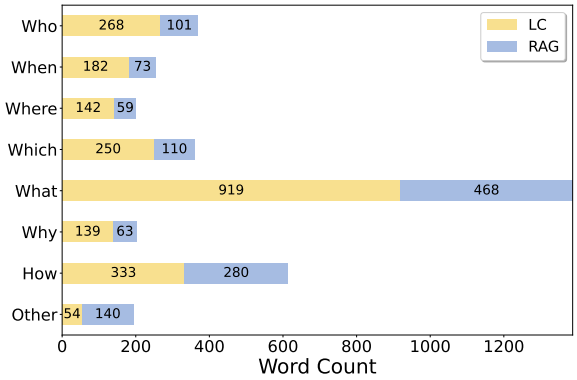


Figure 4: Performance breakdown by question type for LC Only and RAG Only.

In contrast, RAG performs better with dialogue-related sources and achieves comparable performance with papers or reports. The information in these sources is naturally segmented, conversations have turns, and papers and reports have clearly defined sections or subsections, making the retrieval of key segments easier.

Figure 4 shows that LC performs better for fact-based questions such as "Who", "Where", and "Which". These questions often benefit from having all the relevant context available in a dense region close to the answer. RAG, however, is largely comparable to LC for more open-ended questions such as "How", which often require synthesizing information from multiple sources and therefore benefit from retrieval-based approaches.

Furthermore, RAG outperforms LC in the "Other" questions, which consist mainly of general questions that can be answered with "Yes" or "No". We hypothesize that the reason could be due to the training data. Long-context LLMs are more familiar with phrasing of common type questions than general questions. Words like "Who" or "Where" act as keywords for long-context LLMs to search, while retrievers use these keywords not so well.

5 What is Long Context?

Although we have reviewed 9 studies that either directly or implicitly compare or integrate RAG and Long Context, very few studies clearly define what Long Context is. To this end, we separately interpret the two words 'long' and 'context'.

5.1 Defining "Long"

Out of the 9 studies reviewed earlier, only 2 studies, ChatQA2 and LongBench v2 explicitly define Long Context as greater than 32k and greater than 8k tokens respectively. For other studies, we can only infer their definitions of "long" based on the models and datasets they use. It seems that three studies consider 8k as a minimum requirement for long context, and another three studies set this requirement at 16k. Lastly, OP-RAG regards 128k as long context.

In short, each work defines 'Long Context' based on its own criteria due to the lack of a clear standard. Moreover, as the context windows of language models continue to expand, the terms 'long' and 'short' are relative. For example, 4k tokens are not considered 'long context' in any of the reviewed studies but are extremely long for BERT-base models, which support only 512 tokens. As a result, the definition of 'long' remains ambiguous, leading to inconsistent use of this concept among researchers. In practice, the definition of 'long' is complicated, depending on the context length of latest LLMs, and the length of the documents in targeted domain.

5.2 Defining "Context"

In the English dictionary, 'context' is defined as "the situation within which something happens, and that can help explain it". By this definition, the context of a question is expected to "help explain it", implying that the context should have strong relevance to the question. However, long-context datasets are not always constructed with this principle in mind. The construction of long-context datasets can generally be categorized into two types:

Realistic Long Texts: These datasets originate from sources such as novels, research papers, or other lengthy narratives, exemplified by datasets like NovelQA. Such datasets typically pose challenges that involve reading comprehension and require models to process and synthesize dense information spread across a cohesive, extended text.

Synthetic Long Texts: These datasets are often created by concatenating smaller, query-relevant segments of text, such as Wikipedia-sourced datasets in LongBench. This construction process may involve stitching together Wikipedia excerpts, injecting noise, or combining unrelated passages to simulate a long document.

A critical observation is that realistic long contexts align more closely with reading comprehension tasks, where models primarily absorb and reason over information. Such datasets have high contextual relevance, since the questions are normally based on the documents that users provided. In contrast, synthetic long contexts often resemble factual reasoning tasks, where models retrieve and verify knowledge. Such datasets inherently incorporate a pre-processing step like a RAG pipeline. They can assess the impact of information placement on model performance, such as the lost-in-the-middle phenomenon.

On the other hand, realistic and synthetic long texts can only serve as proxies to reflect context relevance to some extent. The scope of the context is question-dependent and difficult to define clearly.

6 Conclusion

In this paper, we survey existing studies comparing or combining LC and RAG, analyzing why different implementations may result in some conflicts among their insights. Therefore, we present a thorough comparison of LC and RAG approaches by leveraging a diverse set of long context QA datasets. We filtered out questions that could be answered from parametric knowledge, ensuring a fair comparison by focusing on questions that required external context. Along these lines, we have developed a systematic filtering and evaluation process, identified the best retrieval method, and expanded the dataset to provide a statistically significant basis for analysis. The results indicate that LC generally outperforms RAG for tasks involving well-structured, dense contexts, and is better at answering questions requiring specific information. By contrast, RAG demonstrates advantages in handling fragmented information.

Beyond merely presenting the experimental results and findings, we delve deeper into the concept of long context and examine how LC and RAG should be compared. Our discussion aims to ensure that the insights gained are more impactful and applicable to real-world scenarios.

Limitations

While our study provides valuable insights into the comparative strengths and weaknesses of Long Context (LC) and Retrieval-Augmented Generation (RAG) approaches, it is important to acknowledge three limitations that may impact the generalizability and comprehensiveness of the findings:

Our analysis is limited to text-based long contexts, and neglecting other modalities such as audio, video, or multi-modal contexts. The applicability of these insights to non-textual long-context scenarios remains unexplored, which may limit the broader applicability of the findings to multi-modal applications.

Our work focuses on existing papers that compare and combine RAG with long-context LLMs. Therefore, we mainly survey the retrievers and LLMs used in those papers, rather than all available retrievers and long-context LLMs.

Our experiments rely on existing LC and RAG implementations, including specific retrieval methods and strong long-context models. As the field continues to evolve, newer models or retrieval strategies may alter the comparative outcomes. However, our evaluation framework is still applicable to future evaluation.

Ethical Considerations

Advanced Long Context LLMs equipped with strong RAG capabilities could be misused to generate misleading or harmful content, such as fake news or propaganda. Their long-context capability could amplify the scale and believability of such content. Researchers should prioritize safety and transparency in model usage to mitigate the risk.

References

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2024. L-eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14388–14411, Bangkok, Thailand. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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. 2024a. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024b. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *CoRR*, abs/2412.15204.

Victoria Basmova, Yoav Goldberg, and Reut Tsarfaty. 2024. Llms' reading comprehension is affected by parametric knowledge and struggles with hypothetical statements. *CoRR*, abs/2404.06283.

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, and Christopher Hesse et al. 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*.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, and Tao Gui et al. 2024. Internlm2 technical report. *CoRR*, abs/2403.17297.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *CoRR*, abs/2306.15595.

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*, pages 4599–4610, Online. Association for Computational Linguistics.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and Guowei Li

et al. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *CoRR*, abs/2405.04434.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, and Angela Fan et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Weizhi Fei, Xueyan Niu, Pingyi Zhou, Lu Hou, Bo Bai, Lei Deng, and Wei Han. 2024. Extending context window of large language models via semantic compression. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 5169–5181. 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.

Shweta Gupta, Sunita Yadav, and Rajesh Prasad. 2018. Document retrieval using efficient indexing techniques: A review. *Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications*, pages 1745–1764.

Donna Harman. 1992. Relevance feedback revisited. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Copenhagen, Denmark, June 21-24, 1992*, pages 1–10. ACM.

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*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. Financebench: A new benchmark for financial question answering. *CoRR*, abs/2311.11944.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024a. Mixtral of experts. *CoRR*, abs/2401.04088.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. 2024b. Longrag: Enhancing retrieval-augmented generation with long-context llms. *CoRR*, abs/2406.15319.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan Ö. Arik. 2024. Long-context llms meet RAG: overcoming challenges for long inputs in RAG. *CoRR*, abs/2410.05983.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Quinn Leng, Jacob Portes, Sam Havens, Matei Zaharia, and Michael Carbin. 2024. Long context rag performance of large language models. *CoRR*, abs/2411.03538.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval augmented generation or long-context llms? A comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - Industry Track, Miami, Florida, USA, November 12-16, 2024*, pages 881–893. Association for Computational Linguistics.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6385–6400, Singapore. Association for Computational Linguistics.

Jerry Liu. 2022. LlamaIndex. *CoRR*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and Pouya Tafti et al. 2024. Gemma: Open models based on gemini research and technology. *CoRR*, abs/2403.08295.

Erik Nijkamp, Tian Xie, Hiroaki Hayashi, Bo Pang, Congying Xia, Chen Xing, Jesse Vig, Semih Yavuz, Philippe Laban, Ben Krause, Senthil Purushwalkam, Tong Niu, Wojciech Kryscinski, Lidiya Murakhovs'ka, Prafulla Kumar Choubey, Alex Fabbri, Ye Liu, Rui Meng, Lifu Tu, Meghana Bhat, Chien-Sheng Wu, Silvio Savarese, Yingbo Zhou, Shafiq Rayhan Joty, and Caiming Xiong. 2023. Xgen-7b technical report. *CoRR*, abs/2309.03450.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, and Jeff Belgum et al. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and Junyang Lin et al. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, and Julian Schrittwieser et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 12991–13013. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and Brian Fuller et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

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.

Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Qian Wang, and Yue Zhang. 2024a. Novelqa: A benchmark for long-range novel question answering. *CoRR*, abs/2403.12766.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11897–11916. Association for Computational Linguistics.

Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. 2024c. Beyond the limits: A survey of techniques to extend the context length in large language models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 8299–8307. ijcai.org.

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

Peng Xu, Wei Ping, Xianchao Wu, Zihan Liu, Mohammad Shoeybi, and Bryan Catanzaro. 2024a. Chatqa 2: Bridging the gap to proprietary llms in long context and RAG capabilities. *CoRR*, abs/2407.14482.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina

Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and Jianxin Yang et al. 2024. Qwen2 technical report. *CoRR*, abs/2407.10671.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William 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*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Tan Yu, Anbang Xu, and Rama Akkiraju. 2024. In defense of RAG in the era of long-context language models. *CoRR*, abs/2409.01666.

Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, and Lei Zhao et al. 2024. Chatglm: A family of large language models from GLM-130B to GLM-4 all tools. *CoRR*, abs/2406.12793.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. RAFT: adapting language model to domain specific RAG. *CoRR*, abs/2403.10131.

A Related Work

Our primary focus is to evaluate and compare LC and RAG. To this end, we review papers with a similar focus, and provide a detailed analysis of the retrievers and long-context settings they employ.

A.1 Retrievers

Retrievers, as fundamental components of RAG pipelines, focus on identifying and extracting contextually relevant segments of documents. We categorize retrieval strategies into three main approaches: *chunk-based retrieval*, which splits documents into smaller segments and then retrieves those most relevant to a query; *index-based retrieval*, which builds specialized index structures to guide efficient and context-rich lookups; and *summarization-based retrieval*, which leverages hierarchical summaries to capture a document's key information at various levels of abstraction.

Chunk-based Retrieval can be broadly categorized into sparse retrievers and dense retrievers. Sparse retrievers, such as the classic BM25 (Robertson and Zaragoza, 2009), operate on term frequency-based representations of text and rank chunks based on a similarity function, leveraging exact matches and term weighting. With the advent of word embeddings, dense retrievers have gained prominence. These models encode both queries and document chunks into dense vector representations and calculate relevance using similarity metrics, such as cosine similarity.

Since text similarity is often defined by measuring the distance between embeddings, the quality of these embeddings is particularly important. Contriever (Izacard et al., 2022) leverages contrastive learning for training without supervision. By generating synthetic queries and pre-training on unlabeled data, Contriever provides robust retrieval capabilities especially in cross-lingual applications. On a larger scale, BGE-Large (Xiao et al., 2023) employs diverse datasets and sophisticated training methods to outperform previous models on comprehensive benchmarks such as C-MTEB. E5Mistral-7b (Wang et al., 2024b) combines open-source, decoder-only LLMs with synthetic data generation pipelines. With minimal human annotations, the fine-tuning achieves SOTA performance on BEIR and MTEB. Dragon (Lin et al., 2023) also employs data augmentation, including cropping and generative queries, and integrates labels from multiple retrieval sources. This strategy ensures its effec-

tiveness without increasing model complexity. Another method of learning high-quality embeddings is through strong generalization ability from LLMs. For instance, OpenAI embeddings draw upon the GPT-3.5/4 family while Zhipu-embedding-3 leverages the GLM family (Zeng et al., 2024).

Index-based Retrieval requires pre-processing on the documents with more complicated data structures (Gupta et al., 2018). With the development of LLM, Llama-Index (Liu, 2022) was proposed to facilitate interaction between the model and documents more conveniently. The index provides a flexible interface to construct various data structures, known as "indices" that store, organize, and facilitate quick retrieval of context. Once created, these indices can be efficiently queried, guiding the LLM to the most relevant information, improving the accuracy of responses. Some classic indexing methods include tree index which constructs a hierarchical tree from nodes, and knowledge graph index, which builds a knowledge graph with labeled nodes and relationships.

Summarization-based Retrieval is built on top of chunk- and index-based approaches. It provides comprehensive summaries for key points in a document. These summaries available for retrieval.

RAPTOR (Sarthi et al., 2024) improves retrieval by generating recursive summaries of text chunks organized in a tree structure. Instead of retrieving short, contiguous text snippets, RAPTOR clusters text segments, summarizes them at various levels, and forms a hierarchical tree that represents the document's content at different levels of abstraction. This allows retrieval models to extract context at varying levels of detail, improving the ability to handle complex questions that require synthesizing information from multiple parts of the document. Such a summarization-based retrieval method enhances retrieval accuracy for tasks requiring long-range or multi-step reasoning.

A.2 Long-Context LLMs

Many research efforts focus on extending input and output windows to accommodate more context (see Figure 1b), enabling applications such as extended dialogues, large document processing, and complex multimodal tasks. Thus, our analysis focuses on two dimensions: the model capabilities and the context length they can reach.

Model Ability. While most of the models discussed here excel at understanding long documents, many emphasize specialized capabilities. ChatGLM2-6B-32K (Zeng et al., 2024) employs Multi-Query Attention to achieve high reasoning efficiency with low memory usage, making it suitable for tasks requiring deep reasoning. XGen-7B-8K (Nijkamp et al., 2023) enhances long-context conversational understanding and text summarization, enabling coherent and contextually rich dialogues. InternLM-7B-8k (Cai et al., 2024) is optimized for knowledge understanding, reading comprehension, and multilingual translation, supporting diverse linguistic applications. Models like DeepSeek-V2-Chat (DeepSeek-AI et al., 2024), Qwen2-72B-Instruct (Yang et al., 2024), Qwen2.5-72B-Instruct (Qwen et al., 2024), Mixtral-7x8b (Jiang et al., 2024a), and DBRX-Instruct excel in mathematical computations, logical reasoning, and coding, demonstrating strong performance in technical and analytical tasks.

Additionally, Claude-3-Opus, Sonnet, Haiku, Gemini-1.5-flash, and Gemini-1.5-pro (Reid et al., 2024) incorporate multi-modal capabilities, effectively handling both textual and visual information. GLM-4-9B-Chat (Zeng et al., 2024), Mistral-12b-Instruct, and Llama-3.1-Instruct (Dubey et al., 2024) offer robust multilingual abilities, strong instruction-following and multi-turn dialogue capabilities, increasing their utility in a wide range of conversational scenarios. Finally, Claude-2 is notable for low hallucination rate when processing extra-long documents, ensuring high accuracy and reliability in information retrieval and synthesis.

Context Length. As shown in Figure 1b, there is a clear trend of increasing context length in newly released models. Following the categorization approach proposed by ChatQA2 (Xu et al., 2024a), we classify these models into three categories based on their supported context windows: short (up to 4K), long (up to 32K), and ultra-long (more than 32K) context models.

Short context models, such as Llama2-70B and llama2-7B-chat-4k (Touvron et al., 2023), support up to 4K tokens and are typically employed as baselines for retrieval and standard conversational tasks. *Long context* models, including XGen-7B-8K(Nijkamp et al., 2023), InternLM-7B-8k(Cai et al., 2024), Mixtral-7x8b (Jiang et al., 2024a), DBRX-Instruct and Gemma2-9B (Mesnard et al., 2024), offer context windows ranging from 8K to

32K tokens. These are ideal for extended conversations, comprehensive text analysis, and detailed summarization tasks. *Ultra-long context* models extend beyond 32K tokens. For example, Claude-2 provides a 100K token window, while Claude-3-Opus, Sonnet, and Haiku handle up to 200K tokens. GPT-4-Turbo(OpenAI et al., 2023), GPT-4o, and GPT-o1 all support 128K tokens, as do DeepSeek-V2-Chat(DeepSeek-AI et al., 2024), Qwen2-72B-Instruct(Yang et al., 2024), Qwen2.5-72B-Instruct (Qwen et al., 2024), GLM-4-9B-Chat (Zeng et al., 2024), GLM-4-Plus, Mistral-12b-Instruct, and Llama-3.1-Instruct. Notably, Gemini-1.5-flahs and Gemini-1.5-pro(Reid et al., 2024) both support up to an unprecedented 10M tokens. These ultra long-context models enable the processing of exceptionally large documents, complex multimodal tasks, and extensive multi-turn dialogues.

A.3 Comparing & Combining LC and RAG

Since the increase in LLMs' context window lengths, some models can contain the entire document, reducing the need to retrieve on documents. Hence, more studies have begun comparing the performance of long-context LLMs and RAG, as well as investigating ways to combine them. LongBench (Bai et al., 2024a) conducts early comparison experiments on a 4K model with RAG and a 32K model. Xu et al. (2024b) systematically compare LC LLMs and RAG, and proposes their combination. LongRAG (Jiang et al., 2024b) introduces long retrievers and long readers, a successful application of long retrieval units to RAG. ChatQA2 (Xu et al., 2024a) instruction-tunes long-context LLMs to a 128K context window and tests their ability with long-context retrievers. Self-ROUTE (Li et al., 2024) enables the model to select either RAG or LC based on self-reflection to reduce costs. OP-RAG (Yu et al., 2024) preserves the original order of retrieved chunks, and LC LLM meets RAG (Jin et al., 2024) investigates long-context LLMs in RAG systems, proposing retrieval reordering methods. LC RAG Performance of LLM (Leng et al., 2024) evaluates the effectiveness of RAG on long-context LLMs across context lengths from 2K to 2M tokens. Very recently, LongBench is updated to LongBench V2 (Bai et al., 2024b), which tests LLMs on long context comprehension and reasoning with a more realistic and challenging setting.

We summarize the key insights from these papers into three categories: (1) general insights such as chunking strategies, (2) combining the two strategies, and (3) comparing the performance between LC and RAG (see Table 1).

Some papers reach consensus on chunking strategy that, retrieval units should be longer (Jiang et al., 2024b) and the number of chunks should be kept low (Yu et al., 2024). According to (Xu et al., 2024b), selecting the top 5 to 10 chunks typically yields strong performance, while retrieving more than 20 chunks leads to diminished results. LongBench (Bai et al., 2024a) presents a different finding, suggesting that splitting a long context into shorter and more numerous chunks is better. However, at the time of its publication, LLMs generally exhibited weaker long-context capabilities, and the study did not incorporate very long retrieval units (>1000 tokens). Consequently, LongBench's findings are not at odds with the broader consensus.

Nonetheless, these papers present disagreement regarding performance of retrieval on long-context LLMs. For instance, LongBench (Bai et al., 2024a) finds that retrieval helps short-context models but not 7B long-context models. In contrast, Xu et al. (2024b) suggest that RAG improves 70B models across all context lengths, attributing the discrepancy to the difference between model sizes. Similarly, ChatQA2 (Xu et al., 2024a) observes that increasing the context window from 3K to 24K tokens consistently benefits RAG. Notably, LongBench V2 (Bai et al., 2024b) shows that GPT-4o continues to improve in RAG performance even at 128K input, whereas Qwen2.5 and GLM-4-Plus show performance deterioration beyond 32K input. The observations align with findings from (Leng et al., 2024) that RAG for close-source models can improve up to 100K input, whereas performance for some open-source models peaks around 16K tokens. Hence, the varying behaviors might be due to different model size and architecture.

There are even greater discrepancies in the direct comparisons between the two methods. Xu et al. (2024b) claims that long-context models outperform retrieval with short-context models in multi-hop benchmarks. In contrast, ChatQA2 (Xu et al., 2024a) finds that RAG can outperform LC if a sufficient number of top-k chunks are used. Self-ROUTE (Li et al., 2024) fully supports LC, arguing that it outperforms RAG in all benchmarks. Meanwhile, OP-RAG (Yu et al., 2024) defends RAG, demonstrating that efficient retrieval strategies can outperform a brute-force approach of processing extremely long contexts.

The reasons for the differences among these stud-

ies are manifold. For instance, There are three categories of retrieval methods (*i.e.,* chunk-based, index-based, and summarization-based retrieval), but current studies rely predominantly on chunk-based retrieval, leaving room for further optimization. Additionally, evaluation scores often represent weighted averages across different datasets. Because each dataset has distinct characteristics, placing more emphasis on one dataset and less on another can alter the final results. Finally, most existing studies use only a few datasets with around 200 questions each. This small sample size creates greater room for variability and reduces the general reliability of these findings.

B F1 Score Computation

To calculate the $F1$ score, we first convert both the prediction and the reference text into sets of unique tokens. Tokens appearing in both sets count as true positives (TP), tokens present only in the prediction are false positives (FP), and tokens missing from the prediction but in the reference are false negatives (FN). Precision is defined as $\frac{\text{TP}}{\text{TP+FP}}$, recall as $\frac{\text{TP}}{\text{TP+FN}}$, and the F1 score is their harmonic mean:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

Example:

"cat leaps table quickly"(prediction)
"the cat leaps over the table" (reference)

The corresponding sets are:

$$\text{prediction_set} = \{cat, leaps, table, quickly\}$$

$$\text{gold_set} = \{the, cat, leaps, over, table\}.$$

Here, $\{\texttt{cat}, \texttt{leaps}, \texttt{table}\}$ are TP = 3, $\{\texttt{quickly}\}$ is FP = 1, and $\{\texttt{the}, \texttt{over}\}$ are FN = 2. Hence:

$$\text{precision} = \frac{3}{3+1} = 0.75, \quad \text{recall} = \frac{3}{3+2} = 0.60,$$

$$F1 = 2 \times \frac{0.75 \times 0.60}{0.75 + 0.60} = 0.67.$$

C Further Experiment Results

C.1 In-detail Analysis on Loose Evaluation Settings

As a complement to § 4.3, we provide a detailed comparison of the performance of LC and RAG under the loose evaluation settings based on Exact Match (EM) and F1 scores.

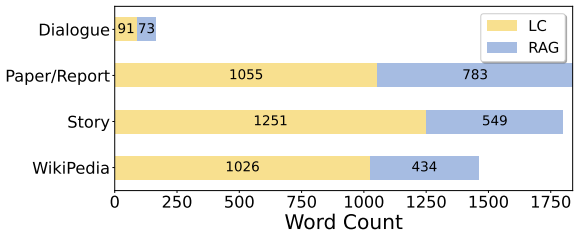


Figure 5: Performance breakdown by knowledge source for LC Better and RAG Better.

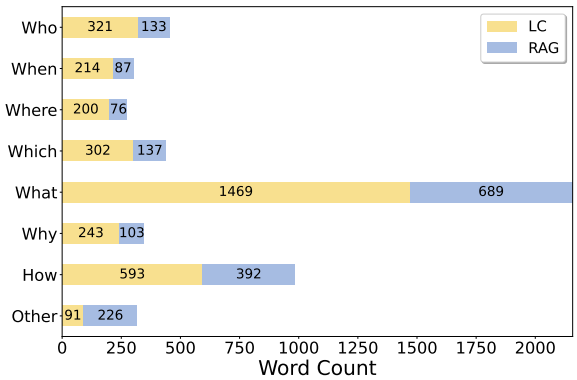


Figure 6: Performance breakdown by question type for LC Better and RAG Better.

As shown in Figure 5, loose evaluation setting reveals similar trends to the strict setting in the performance of LC and RAG on different knowledge sources. LC outperforms RAG for structured sources like Wikipedia, course websites, and papers/reports, where having complete context is advantageous. This trend is consistent in both evaluation settings. However, RAG performs better with dialogue-based and story-based knowledge sources, where the information is fragmented. The loose evaluation, with the inclusion of F1 scores, shows a slight improvement for RAG in these cases, as partial answers are rewarded more, but the overall trend remains the same.

Figure 6 highlights the performance of LC and RAG across different question types. For fact-based questions (e.g., “Who”, “Where”, “Which”), LC continues to outperform RAG in both evaluation settings, as these questions benefit from having complete, uninterrupted context. For open-ended questions (e.g., “How”, “Why”), RAG shows comparable performance to LC in both settings. The loose evaluation, however, slightly favors RAG due to its ability to synthesize information from multiple sources, as F1 scoring acknowledges partial correctness. In the case of "Other" questions (sim-

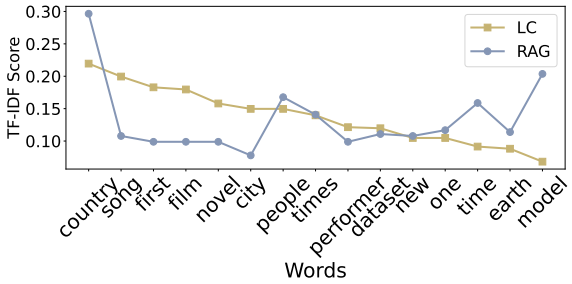


Figure 7: Top 15 Words based on TF-IDF Score for LC Only vs. RAG Only.

ple "Yes" or "No" questions), RAG significantly outperforms LC in both evaluation settings, but the advantage is more pronounced in the loose evaluation. The inclusion of F1 scores helps RAG capture partial successes that would be penalized under strict EM-only scoring.

Overall, the figures illustrate that the performance patterns of LC and RAG remain largely consistent across both strict and loose evaluation settings. The key difference is that RAG gains a slight performance boost in the loose evaluation.

C.2 Word Frequency Visualization

To better understand the scenarios that LC and RAG each excels at, we visualize the word frequencies by their TF-IDF scores, plotted in Figure 7. The TF-IDF scores were calculated from questions in the datasets where either LC or RAG produced correct answers exclusively. Specifically, all questions from each dataset are concatenated and treated as a single document for this analysis, meaning that the TF-IDF scores primarily reflect the term frequency within each dataset. Stopwords are removed and not shown in the plot.

Figure 7 presents the top 15 words that appear most frequently combined in both LC only and RAG only questions. Words such as 'song', 'film', and 'novel' have higher TF-IDF scores for LC, suggesting that LC performs better with narrative topics. Conversely, words like 'country', 'dataset', and 'model' have higher scores for RAG, indicating its strength in retrieving information on technical or data-oriented topics. This analysis underscores the complementary strengths and limitations of LC and RAG in handling different types of questions.

C.3 Impact of Generation Model in RAG

We now evaluate the impact of different generation models on RAG's performance. Table 6 shows the results of using GPT-4o and GPT-4-Turbo as the generator with three retrievers (BM25, Tree Index, RAPTOR), each of which represents one retriever type. The results indicate that the performance of different generation models remains largely consistent regardless of the retriever used. RAPTOR performs the best across both generation models, though there is a slight decrease in performance when using GPT-4-Turbo compared to GPT-4o.

Retriever	Model	Correct (%)	RAG Only	RAG Better
BM25	GPT-4o	319 (20.4)	50	141
	GPT-4-Turbo	310 (19.8)	51	152
Tree-Index	GPT-4o	470 (30.1)	82	234
	GPT-4-Turbo	458 (29.3)	81	229
RAPTOR	GPT-4o	**602 (38.5)**	97	258
	GPT-4-Turbo	**589 (37.7)**	99	295

Table 6: Results of using different generation models

While GPT-4o slightly outperforms GPT-4-Turbo across all retrievers, the differences are marginal. This implies that both generation models are capable of generating high-quality responses, and the choice between them may depend more on other factors such as efficiency or resource availability. The consistency across retrievers also demonstrates that the retrieval method plays a larger role in determining overall performance than the specific generation model used. We will report the results from other models and the experiment is in progress.

C.4 Case Study

For a deeper understanding of the difference between LC and RAG, we conduct a case study to analyze the frequent errors from each method, and present them in Tables 7 and 8. We manually examine the questions that only RAG made mistakes, and those only LC made mistakes.

The most frequent mistake made by RAG is its failure to retrieve the relevant context, leading to its refusal to answer the question. As shown in Table 7, the model correctly identifies that Anthony Upko was formerly involved in the government of Nigeria but fails to retrieve the debt-to-GDP ratio as part of the context. This retrieval failure can arise due to two possible reasons: the retriever might fail to locate the relevant sentences from documents, or the sentences may be split across two chunks, with the debt-to-GDP ratio lacking a clear subject. Interestingly, when provided with the same prompt, LC rarely reports a lack of context, suggesting its robustness in handling such cases.

Another error made by RAG is misinterpreting

Question: What is the debt-to-GDP ratio of the country where Anthony Upko was formerly involved in the government? **Wrong Answer:** The context does not provide the debt-to-GDP ratio for Nigeria. **Gold:** 11 percent **Relevant Sents:** 1. Nigeria is the world's 20th largest economy ... the debt-to-GDP ratio is only 11 percent. 2. Anthony Ukpo was Minister of Information and Culture, and then Governor of Rivers State, Nigeria.
Question: When is the performer of song Swing Down Sweet Chariot 's birthday? **Wrong Answer:** May 8, 1940 **Gold:** January 8, 1935 **Relevant Sents:** 1. Swing Down Sweet Chariot is a traditional song ... recorded by Elvis Presley. 2. Elvis Aaron Presley (January 8, 1935 - August 16, 1977), also known as ...

Table 7: Examples cases where RAG made mistakes

Question: Do the tweets come from a specific region? **Wrong Answer:** Yes, the tweets come from 16 different countries. **Gold:** No **Relevant Sents:** This helped us narrow down our query space to 16 countries.
Question: Where did Valancourt lose his wealth? **Wrong Answer:** In Gambling. **Gold:** Paris **Relevant Sents:** Returning to her aunt's estate, Emily learns that Valancourt has gone to Paris and lost his wealth.

Table 8: Examples representing common cases where only RAG answers correctly

partial context. In the second example, where RAG incorrectly answered the birthday, the model retrieved May 8, 1940, instead of the correct date, January 8, 1935. This occurred because the sentence 'Swing Down Sweet Chariot is a traditional song ... recorded by Elvis Presley' spans too long, creating ambiguity in linking the birthday to the correct person. This type of retrieval failure highlights a core limitation: RAG relies heavily on retrieving continuous text spans, and any fragmentation or overly long context can lead to an incomplete understanding. In contrast, LC tends to provide more holistic answers when processing longer contexts directly, as it bypasses the dependency on a retrieval module.

Wrong answers by LC are often caused by question misinterpretation. For instance, as shown in Table 8, when asked whether the tweets come from a specific region, LC answers 'yes', referencing that the tweets originate from 16 countries. It fails to interpret the relationship between 'a specific region' and '16 different countries'. In another example, when asked 'where' Valancourt lost his wealth, the model identifies the correct sentence but answers 'how' instead of 'where'. These examples highlight that LC sometimes struggles to align its semantic understanding with the required level of specificity or perspective, resulting in answers that are related but not addressing the question's intent. In both cases, the LLMs are able to locate the related texts from the documents, but the reasoning ability might be affected by the noise.

D Further Discussion

D.1 How to Compare or Combine LC & RAG?

The lack of a clear definition for long context also indicates the absence of a coherent framework for comparing or combining LC and RAG. We propose such a framework by examining three key perspectives: context length, context relevance, and experiment design.

Context Length. From the model's perspective, context length refers to the maximum number of tokens a model can process. From the dataset's perspective, it denotes the amount of text provided with a question. In synthetic datasets, context length is flexible, but this introduces a trade-off between length and relevance. Adding irrelevant information as context may help to test a model's robustness to noise, but such testing may not represent real-world use cases. Therefore, any framework for comparing LC and RAG should clearly define what is considered 'long', while indicating whether this length criterion originates from the model's capabilities, the dataset's design, or both.

Context Relevance. An evaluation framework must also address the relevance of the text provided as input to the model. It is crucial to distinguish between realistic long contexts and synthetic long contexts. When benchmarks include both types, separate evaluations are necessary, as synthetic contexts often have low relevance and may not accurately reflect real-world scenarios.

Interestingly, the construction of synthetic long contexts often mirrors RAG pipelines. Providing an entire curated text to an LLM as context essentially

represents a 'long context RAG' approach, given that such text is assembled during dataset creation. Further chunking can introduce biases against RAG by disrupting the continuity of information within each piece.

Additionally, many benchmarks categorize tasks as 'single-doc' or 'multi-doc' based on whether the text originates from a single source or multiple documents. While convenient, this categorization does not perfectly align with 'realistic' or 'synthetic' contexts. A single document may sometimes be artificially composed of smaller fragments, while a multi-sourced document might involve highly relevant sources, such as a group of research papers discussing the same problem.

The key issue remains determining to what extent the context provided as input to LLMs contains sufficient and relevant content to answer the question, without introducing unnecessary or unrelated information.

Experiment Settings. When investigating LC and RAG, the experimental objectives can be broadly grouped into two categories: comparison and combination.

Short RAG v.s. Long Single Input: one might compare a short-context RAG pipeline against a long-context single-input setup, analyzing both performance and computational cost. This provides insights into the trade-off between running an extra retrieval pipeline for shorter contexts versus allowing the model to process a larger uninterrupted text.

Long RAG v.s. Long Single Input: One may also compare a long-context RAG pipeline with a long-context single-input approach. Here, the goal is to see whether chunking or filtering more relevant content through retrieval can outperform or complement a fully integrated long-context approach by truncating exceptionally long documents.

In the first setting, the retrieval pipeline naturally reduces the number of tokens. In the second setting, the context length remains the same for both methods, with the only difference being how the text is processed.

RAG over Increasing Context: Another possible goal is understanding how RAG performance changes with increasing context lengths. In this scenario, the "LC" refers specifically to how many tokens a model can handle. This line of work can reveal how well RAG pipelines scale when models absorb increasingly larger inputs.

On the other hand, findings from evaluations often serve as guidelines for settings that address real-world problems. In this sense, RAG and LC may complement each other in real-world settings, depending on the characteristics of the data source and the types of questions to be answered.

D.2 Revisiting All Studies

Based on the earlier discussion, the exploration of LC and RAG methods in LLMs highlights some critical challenges that researchers often overlook.

Trade-off between Context Length and Relevance. Many studies hesitate between using flexible synthetic context with noisy concatenated contexts, or realistic context with dense information but less availability. Among the 9 studies, 6 select synthetic context as part of the datasets. Our own evaluation has also selected synthetic context datasets, but we consider the influence of synthetic long context and separately evaluate their results by context source; e.g. a Wikipedia source with manual noises represents low context relevance.

Several studies have attempted to address this challenge. LongBench recently updated v2 which collects only realistic data. Despite a smaller scale, LongBench v2 shows substantial improvement in context relevance compared to its first version. LongRAG retrieves from a massive corpus for all questions, instead of assigning one context to each question. This method avoids retrieving from a synthetic long context and is hence recommendable.

Diversity in Retrieval Mechanisms. In the comparison of RAG and LC, RAG is often underrepresented due to an over-reliance on traditional retrieval strategies. Among the 9 studies, 5 experiment with different retrievers, only 2 try different chunking sizes, and none consider any retrieval method beyond chunk-based retrievers. Although we experiment with index-based and summarization-based retrievers, we cannot promise that our selected method outperforms all retrieval strategies.

For investigating RAG performance over increasing context, some studies propose their own strategies for chunking and placing RAG. OP-RAG proposes preserving the original order of chunks from the context, while LC LLM-RAG proposes placing higher-scored chunks at the front and back. In addition to more advanced retrievers, certain information retrieval (IR) (Manning et al., 2008) techniques like relevance feedback (Harman, 1992) or query expansion (Carpineto and Romano, 2012)

might further enhance RAG performance, yet these have been overlooked in existing frameworks.

Computational Cost. Most existing studies test on 6 to 8 datasets, and it becomes increasingly expensive to conduct experiments on too many models. This is especially the case when new long-context LLMs are being released at a very fast pace. Hence, any work might be questioned because the experiment results are only applicable to one or a few models. Among all works, LC RAG Performance includes the largest number of models (20). While their efforts are remarkable, they only experiment on 3 datasets. FinanceBench (Islam et al., 2023) looks at finance domain, Databricks DocsQA is based on Databricks platform, and NQ as shown table 2 as a very low rate of requiring external knowledge. This is not meant as criticism but rather to show the trade-off between testing many models and having a comprehensive benchmark.