Letriever: Leveraging LLMs as Context Retrievers with Reasoning Capabilities

Anonymous EMNLP submission

Abstract

Large Language Models (LLMs) have demonstrated outstanding performance on Question Answering (QA) tasks. However, they face significant challenges due to difficulties in effectively utilizing lengthy inputs, resulting in irrelevant responses. Retrieval-Augmented Generation (RAG) frameworks have been employed to address this issue. However, they remain limited by prioritizing superficial lexical overlaps, leading to suboptimal context selection. In this study, we propose ***Letriever***, which replaces the traditional embedding-based retriever with an LLM-based retriever. By leveraging the advanced comprehension capabilities of LLMs, *Letriever* enhances retrieval precision and answer accuracy across diverse QA benchmarks. Our findings highlight the potential of LLMs to transform retrieval mechanisms in QA systems.

1 Introduction

The growing complexity of real-world QA tasks highlights the need for methods that can effectively process and reason over long, information-rich contexts. With the enhanced token capacity, Large Language Models (LLMs) (Achiam et al., 2023; Anthropic, 2024) have shown impressive performance across various QA tasks. However, studies (Shi et al., 2023) reveal that excessively long contexts can introduce noise, making it challenging for models to accurately identify and reference relevant information.

Retrieval-Augmented Generation (RAG) is commonly employed to address this issue. By leveraging semantic search methods such as cosine similarity, RAG filters out irrelevant noise by assuming that the most similar content is also the most relevant. The limitation of this approach is that it may overlook semantically relevant information expressed in different ways, leading to inaccuracies in evidence retrieval and downstream answer generation.

To address this issue, we propose *Letriever*, which replaces the traditional embedding-based retriever with an LLM-based model. Our method leverages the advanced natural language reasoning capabilities of LLMs in an end-to-end manner, where they function as both retrievers and generators. Unlike traditional RAG, our method can understand complex natural language information and ensure robust performance by reducing dependency on hyperparameters such as top-k selection.

We summarize our contributions as follows:

- We propose an LLM-based retriever approach that replaces traditional cosine similarity-based semantic search in Retrieval-Augmented Generation (RAG).
- Through experiments on diverse QA datasets, we demonstrate that LLMs' contextual understanding enables more nuanced semantic representations, leading to higher performance across a variety of QA tasks compared to traditional semantic search.
- We introduce no_k hyperparameter, which leverages the reasoning capabilities of LLMs to flexibly select a various number of contexts based on the query. This eliminates the need to tune the top-k hyperparameter for each dataset.

2 Related Work

2.1 Lost in the Middle problem

Addressing long-context processing has been an active area of research. Several studies have proposed improvements to the attention mechanism, such as Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2020), and Performer (Choromanski et al., 2020). Others have explored incorporating recurrence into the model architecture,

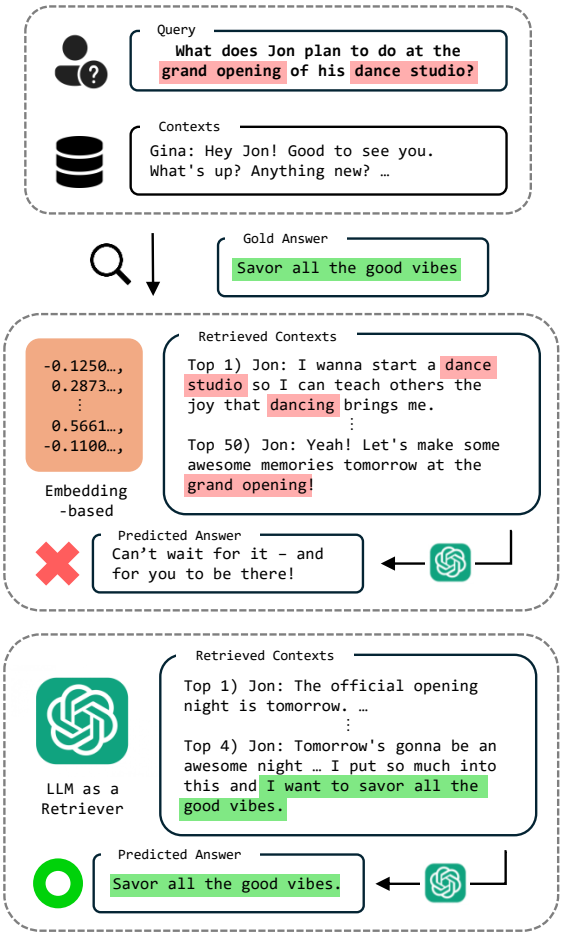


Figure 1: **Comparison between embedding-based and LLM-based retrieval.**

including Transformer-XL (Dai, 2019) and Compressive Transformer (Rae et al., 2019).

However, recent studies suggest that merely expanding the context length is not enough to improve model performance. Longer contexts can lead to the loss of relevant information, especially in the middle of the input, a phenomenon called as the Lost in the Middle problem (Liu et al., 2024b). Moreover, the inclusion of unnecessary or irrelevant information in long contexts can significantly degrade model performance (Shi et al., 2023). How to make effective use of the extended context capabilities of LLMs remains an open question.

2.2 Retrieval-Augmented Generation

As one of the practical solutions to address the Lost in the Middle problem, Retrieval-Augmented Generation (RAG) technique is often employed. RAG is a paradigm to enhance generative models by retrieving relevant information from external knowledge sources and using it as context (Lewis et al., 2020). This approach addresses issues in LLMs, such as hallucination and outdated knowledge by grounding their outputs in external information.

Despite its advantages, RAG faces several challenges. As illustrated in Figure 1, embedding-based retrieval often prioritizes contexts containing exact query terms, overlooking logically relevant contexts without those terms. In contrast, LLMs can retrieve contexts that logically support the query even if they lack query terms. A detailed case study is provided in Appendix B.1.

Another challenge lies in setting the top-k, or the number of contexts to retrieve. Embedding-based retrievers require a fixed top-k, but the optimal number can vary by domains or chunk size. A large top-k may introduce noise, while a small top-k risks missing critical information. While similarity thresholds (Radeva et al., 2024) can dynamically adjust retrieval, they often require fine-tuning for individual data points.

To address these challenges, recent efforts incorporate query rewriting (Ye et al., 2023), adaptive search (Wang et al., 2023b; Jeong et al., 2024), verification (Li et al., 2023), and self-reflection (Asai et al., 2023; Li et al., 2024). We address these challenges by directly employing LLMs as retrievers. This approach retrieves logically necessary contexts without lexical overlaps and introduces a *no_k* setting that does not require a top-k hyperparameter.

2.3 Leveraging LLMs for QA tasks

There have been numerous efforts to leverage language models' reasoning capabilities to improve performance on QA tasks. Representative methods include prompt engineering techniques such as chain-of-thought (CoT) prompting (Wei et al., 2022) and few-shot prompting with GPT-3 (Brown et al., 2020), which enhance reasoning ability without additional training.

However, as QA tasks have evolved, the need to incorporate larger external knowledge sources has become increasingly important. In addition to RAG, reranking methods (Glass et al., 2022) have been proposed that reorder documents retrieved via embedding-based approaches to improve relevance.

Despite these advances, embedding-based retrieval methods often suffer from the limitation that retrieved documents are only semantically similar to the query, potentially missing important contextual information. Moreover, if the initially retrieved

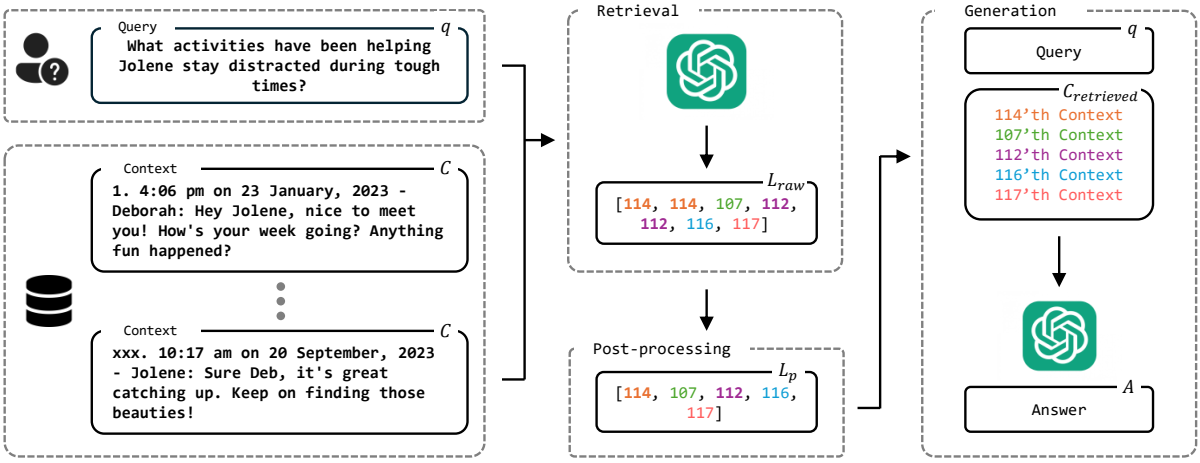


Figure 2: **Framework of Letriever.** There are three stages. **1. Retrieval:** We let LLM retrieve indices of relevant contexts to the question. **2. Post-processing:** We remove the duplicates and preserve the order. **3. Generation:** We replace the indices with the original contexts. Finally, LLM answers to the query with the retrieved contexts.

documents are unrelated to the query, reranking cannot sufficiently improve final performance.

To capture richer textual knowledge, LM-based retrieval approaches have been explored. In REALM (Guu et al., 2020), the language model itself is used to model the distribution over relevant documents given a query, p(z|x), and subsequently generate an answer y based on z and x. Letriever follows a similar idea but performs retrieval solely through LLM inference, effectively extracting contextually relevant passages without additional training.

3 Letriever

This section outlines Letriever, which utilizes LLMs as retrievers in question answering tasks. Our approach consists of three stages: Retrieval, Post-processing, and Generation. The overall framework is illustrated in Figure 2.

3.1 Retrieval

We designed the retrieval phase to be as simple as possible to investigate the ability of LLM to retrieve contexts. We simply provided all the contexts C to the LLM, which are segmented into sentences. Then we instructed with instruction I_r to retrieve k most important contexts for answering the question q. As a result, LLM responds a Python list $L_{\text{raw}} = \{i_1, i_2, \ldots, i_n\}$ containing the indices of the retrieved context to ensure that the original sentences are not modified. Note that the number of retrieved indices n can be different from k. The process can be written as follows:

$$L_{\text{raw}} = LLM(I_r; C; q; k),$$

In contrast to traditional methods, this approach does not retrieve contexts based on embedding similarity with the query. Instead, it relies on the expectation that the LLM can identify relevant contexts that embedding similarity alone may fail to capture.

We provide the prompt used in the retrieval phase in Figure 3. The LLM is instructed to select the relevant contexts that can help answer the question and output them as a Python list of indices.

```
Select {k} important contexts from CONTEXT. Important contexts are those that can help answer the QUESTION. Provide the selected contexts' positions (indices) in a Python list. Provide only the indices as your response. Assume that the index starts from 0. The indices must also be less than {context_len}. Output as a list.

## CONTEXT
{context}
## QUESTION
{question}
```

Figure 3: **Prompt for retrieval stage in Letriever framework.** For $k = no\_k$ setting, we simply do not provide $\{k\}$ to LLM, allowing the LLM to determine it dynamically.

3.2 Post-processing

After receiving the list of indices from the LLM, we further post-processed it. The LLM occasionally included duplicate indices in its response. We removed these duplicate indices. Additionally, the list of indices provided by the LLM was in no particular order; it was neither ascending nor descending. This suggests that the LLM arranged the indices based on their importance in answering the given question. We did not rearrange these indices; instead, we maintained the original order provided by the LLM. As a result, we get a final list of indices of relevant contexts L_p.

3.3 Generation

The generation phase is the same as the traditional embedding-based RAG method. We extract the retrieved contexts $C_{\text{retrieved}}$ by replacing the post-processed indices L_p with the original contexts. Given the retrieved contexts $C_{\text{retrieved}}$ and instruction I_g, a generation model finally generates an answer A based on the question q.

$$C_{\text{retrieved}} = \{C[l] \mid l \in L_p\},$$
$$A = LLM(I_g; C_{\text{retrieved}}; q)$$

4 Experiments

4.1 Datasets

LoCoMoQA (Maharana et al., 2024)**.** LoCoMo is a dataset of very long-term conversations with multi-sessions. We use 1,540 QA pairs in this dataset, excluding adversarial questions that a generation model should answer as 'unanswerable' because the evidence is absent, making them irrelevant to the performance of retrieval methods. These QA pairs include single-hop, multi-hop, temporal, and open-domain questions. In addition, we use two types of retrieval units in the dataset: dialogue and observation, the latter of which refers to information observed in the dialogue history (e.g., 'Caroline attended an LGBTQ support group recently and found the transgender stories inspiring').

QASPER (Dasigi et al., 2021)**.** QASPER is a dataset for question answering on scientific research papers. It consists of 5,049 questions over 1,585 Natural Language Processing papers. We conducted experiments using a dataset of 1,155 QA pairs, excluding unanswerable questions. Each question in the dataset was associated with multiple annotator-provided answers. To evaluate model performance, we calculated scores for all available answers and adopted the maximum score for each question as the final metric.

SQuAD 2.0 (Rajpurkar et al., 2018)**.** This dataset is designed to evaluate reading comprehension and question answering performance. It consists of multiple paragraphs per topic, each containing several question-answer pairs. Since individual paragraphs average around five sentences and do not provide long contexts, multiple paragraphs under the same topic were concatenated into a single context. One question-answer pair was extracted from each paragraph to create the QA dataset. The dev dataset was used, resulting in 4,750 question-answer pairs for experimentation, excluding unanswerable questions.

4.2 Evaluation Metrics

Answer Prediction. We report F1 and ROUGE-L (Lin, 2004) scores for abstractive QA tasks, where the generative model is required to rephrase or summarize relevant information (e.g., LoCoMoQA and QASPER). For the extractive QA task, where the generative model is required to answer by identifying the specific span of text directly from the context (e.g., SQuAD), we report F1 and Exact Match (EM) scores.

Evidence Retrieval. Using the gold evidence labeled in the LoCoMoQA and QASPER datasets, we evaluate evidence retrieval performance based on Precision, Recall, and F1. High Recall indicates that the model successfully retrieves a large portion of the gold evidence context, while high Precision suggests that the retrieved context contains minimal noise. The F1 score provides a balance between these two metrics.

4.3 Experimental Setup

Baselines. We utilize two types of baseline retrieval methods: (1) **Full Context** inputs all contexts to a generation model. Their length is within token limit of the model. (2) **Embedding-based retrievers** retrieve relevant contexts from a vector database by calculating similarity scores between embedded contexts and the question. We employ *DRAGON* (Lin et al., 2023), *E5*$_{\textit{mistral-7b}}$ (Wang et al., 2023a), and *openai-embedding* which is `text-embedding-3-large`[1].

[1]https://platform.openai.com/docs/guides/embeddings

Dataset	Retrieval Method	Answer Prediction		Evidence Retrieval		
		F1	ROUGE-L	Recall	Precision	F1
LoCoMoQA (Dialogue)	Full Context	39.1	38.5	**100.0**	0.3	0.6
	DRAGON	45.1	44.2	81.8	4.5	8.5
	$E5_{\text{mistral-7b}}$	45.2	44.3	87.1	2.4	4.7
	openai-embedding	47.3	46.6	77.6	10.3	18.2
	Letriever (gpt-4o-mini)	48.7	48.0	68.2	**46.1**	**55.0**
	Letriever (DeepSeek-V3)	**52.2**	**51.3**	75.7	43.0	54.8
LoCoMoQA (Observation)	Full Context	28.7	27.7	**100.0**	0.5	1.0
	DRAGON	41.0	40.0	61.8	16.5	26.0
	$E5_{\text{mistral-7b}}$	40.7	39.7	65.8	9.1	16.0
	openai-embedding	41.8	40.9	63.2	17.0	26.8
	Letriever (gpt-4o-mini)	42.9	41.9	56.9	41.4	47.9
	Letriever (DeepSeek-V3)	**44.4**	**43.5**	58.9	**46.0**	**51.7**
QASPER	Full Context	47.9	46.5	**100.0**	5.3	9.7
	DRAGON	42.9	41.2	88.2	13.6	21.9
	$E5_{\text{mistral-7b}}$	45.2	43.1	92.0	12.0	19.8
	openai-embedding	43.8	42.0	91.1	14.0	22.6
	Letriever (gpt-4o-mini)	48.8	47.1	76.6	35.4	39.1
	Letriever (DeepSeek-V3)	**50.5**	**48.9**	63.5	**63.1**	**52.2**

Table 1: **Abstractive question answering performance on the LoCoMoQA and QASPER datasets**. The best performance is marked in **bold**. Results are based on F1-score, ROUGE-L metric for answer prediction, and precision, recall, and F1 scores for evidence retrieval performance.

Retrieval Model. We adopt two base models as retrievers to validate that the framework works with various LLMs using the same prompt. Specifically, we employed `gpt-4o-mini`[2] whose context length is up to 128K tokens, and `DeepSeek-V3` (Liu et al., 2024a) whose context length is up to 64K tokens.

Generation Model. To evaluate retrieval performance, we used a fixed generation model across all baselines and our proposed method. Specifically, we employed `gpt-4o-mini`.

Top-k. We evaluate the baselines and our method with top-k settings of 5, 10, 25, and 50. We also include a *no_k* setting, where the LLM dynamically determines the number of contexts to retrieve. This setting is unavailable to embedding-based retrievers. Retrieval is performed at the sentence level.

4.4 Results and Discussions

Table 1 presents the results for abstractive QA, while Table 2 presents the results for extractive QA. For clarity, the results for each retrieval method in the table were obtained using the specific top-k setting that yielded the best answer prediction performance for that method and dataset. The detailed results with all top-k settings are presented in Appendix A.

Dataset	Retrieval Method	F1	EM
SQuAD	Full Context	63.0	31.7
	DRAGON	75.4	44.6
	$E5_{\text{mistral-7b}}$	74.2	44.3
	openai-embedding	74.9	41.7
	Letriever (gpt-4o-mini)	77.4	44.3
	Letriever (DeepSeek-V3)	**82.7**	**51.8**

Table 2: **Extractive question answering performance on the SQuAD2.0 dataset**. The best performance is marked in **bold**. Results are based on F1-score and EM metric for answer prediction

As Table 1 and Table 2 show, Letriever with `DeepSeek-V3` achieves the best performance across both abstractive and extractive datasets, with F1 scores of 52.2 for LoCoMoQA(Dialogue), 44.4 for LoCoMoQA(Observation), 50.5 for QASPER, and 82.7 for SQuAD.

The Full Context method includes all required

[2]https://platform.openai.com/docs/models#gpt-4o-mini

Method	Answer Prediction (F1)			
	Lcm (Dia.)	**Lcm (Obs.)**	**Qasp**	**Sqd**
Letriever (Ours)	37.8	**35.7**	42.8	77.8
w/ s.o.	36.1	33.6	43.0	76.3
w/ k.d.	**37.9**	**35.7**	42.5	**78.0**
w/ s.o. & k.d.	35.6	33.9	**43.2**	76.6

Table 3: **Ablation Study on Post-processing.** *s.o.* denotes sorting order, and *k.d.* denotes keeping duplicates.

	Lcm (Dia.)	**Lcm (Obs.)**	**Qasp**	**Sqd**
no_k's Rank (among *k*s)	3rd	3rd	1st	1st
Δ to 1st (among *k*s)	-0.4	-0.7	–	–
Δ to Best Baseline	$+4.5$	$+1.9$	$+2.6$	$+7.3$

Table 4: **Performance comparison (F1 score) of the *no_k* configuration across the datasets.** The table shows *no_k*'s rank among various hyperparameter settings (*k*s) on `DeepSeek-V3`, the performance gap to the top-performing setting, and the improvement over the best baseline model.

contexts, achieving 100% recall in retrieval accuracy. Despite containing all necessary evidence, it produced the poorest answer prediction performance, except on the QASPER dataset. This indicates that while LLMs can handle long contexts, their ability to utilize them effectively remains limited. Moreover, longer inputs often generate longer outputs, which can hinder performance in QA tasks requiring concise answers. A related case study is provided in Appendix B.2.

A similar tendency is observed in the performance of embedding-based retrievers. They consistently achieved higher Recall than our method. As discussed in Appendix C, this is because LLM-based retrieval typically retrieves fewer contexts than the specified top-k. Since Recall reflects the proportion of evidence included in the retrieved content, retrieving more contexts generally leads to higher Recall. However, the answer prediction results reveal that higher Recall does not always translate to better performance. This highlights the importance of balancing sufficient retrieval of relevant content with minimizing noise in the process.

Additionally, for the QASPER dataset, Table 1 shows that the Full Context method outperformed all baseline embedding-based retrievers but underperformed compared to our LLM-based retriever approach. These findings suggest that while RAG methods help reduce computational costs, they may fall short on certain datasets. In contrast, LLMs demonstrate strong potential for retrieving relevant contexts based on the query.

5 Analysis

5.1 Ablation Study

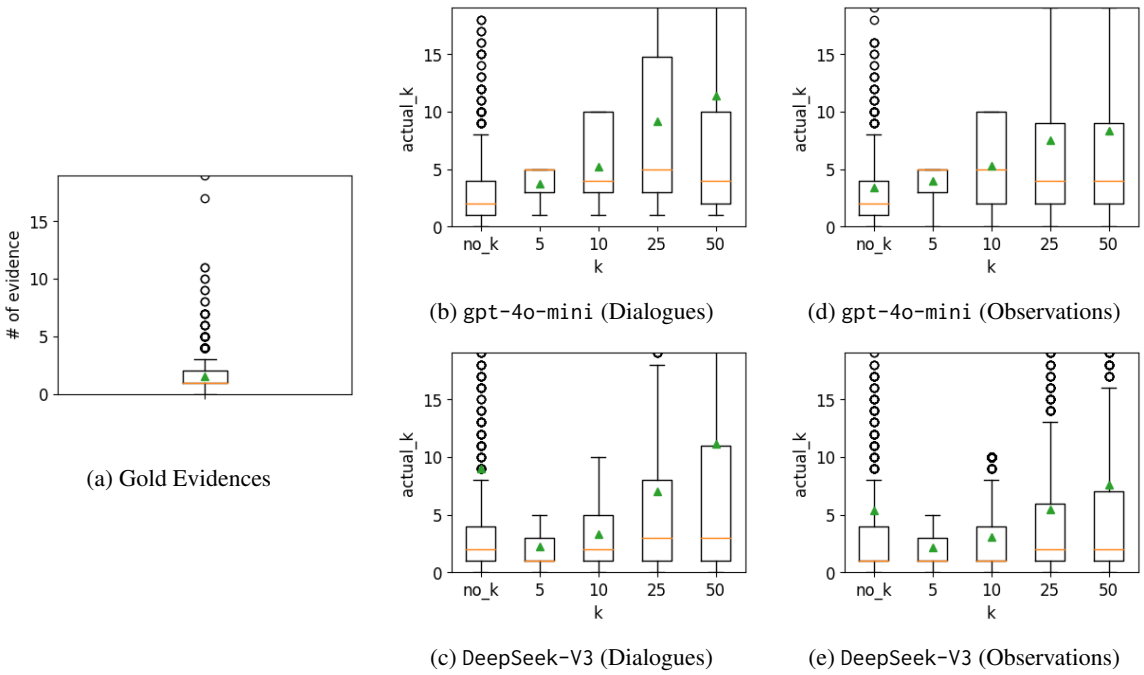
Table 3 represents the answer prediction performance under post-processing ablation. To better assess the impact of each ablation on performance, samples where post-processing did not alter the results were excluded. The ablation results are based on only for `gpt-4o-mini` as a retriever. `DeepSeek-V3` was excluded because the number of samples where post-processing had effect on the results was significantly below 100 per dataset, making the analysis unreliable.

Overall, except for QASPER dataset, preserving both order and duplicates achieved the optimal answer prediction scores. This indicates that LLM retrieves contexts in an order that results in better answer predictions and tends to duplicate important contexts. However, we removed duplicates in the main experiment because the baseline models cannot handle duplicates.

5.2 Effectiveness of the *no_k* Hyperparameter

Based on Table 4, the *no_k* setting demonstrates consistently strong performance across datasets, achieving an average rank of 3rd or better among five different hyperparameter *k* settings. This indicates that, rather than performing extensive and potentially costly hyperparameter tuning, adopting the *no_k* configuration is a cost-effective choice with competitive results.

Even in cases where *no_k* does not rank first, the performance gap to the top-ranked setting is minimal. Furthermore, when compared to the best baseline outside the current hyperparameter search, *no_k* consistently outperforms by notable margins. This shows that *no_k* not only holds its own among optimized settings but also delivers superior results compared to previously established baselines.



(a) Gold Evidences

(b) `gpt-4o-mini` (Dialogues)

(d) `gpt-4o-mini` (Observations)

(c) `DeepSeek-V3` (Dialogues)

(e) `DeepSeek-V3` (Observations)

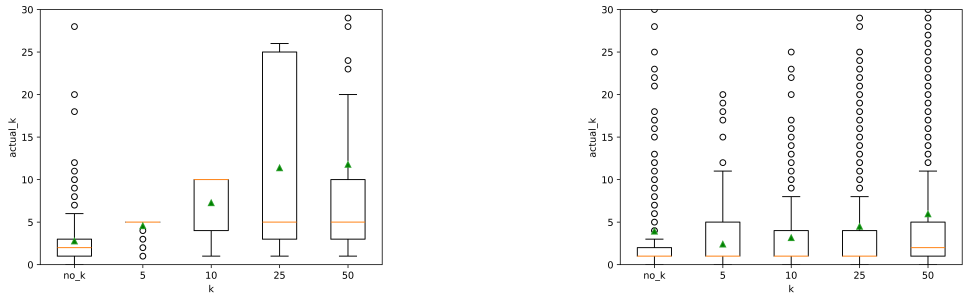
Figure 4: **Comparison of evidence count distribution in the LoCoMoQA dataset**; **(a)**: the number of gold evidences, **(b, c)**: the number of evidences retrieved by LLM where the retrieval unit is dialogue, **(d, e)**: the number of evidences retrieved by LLM where the retrieval unit is observation. Among all of the k settings, no_k, which does not instruct the number of k, demonstrated the closest behavior in the perspective of outliers. For reference, the green triangle represents the average, and the orange line represents the median.

5.3 A Distributional Comparison with Gold Evidences

Figure 4 compares the distribution of gold-labeled evidences in the LoCoMoQA dataset with the evidences retrieved by the LLM. The gold evidence distribution suggests that the optimal number of evidences varies, even across questions within the same dataset. Traditional embedding-based retrievers, which use a fixed top-k setting, cannot replicate this variability, often leading to either the inclusion of noise or the omission of necessary evidence.

In contrast, LLMs can retrieve a variable number of contexts. While the number depends on the top-k setting, the no_k setting shows the closest behavior in the perspective of outliers. Moreover, according to Appendix A, since the no_k setting achieved decent Precision scores, it suggests that LLMs has potential to dynamically select appropriate number of contexts based on the query without the need to set top-k hyperparameter.

Figure 5 shows the distribution of the number of evidences in the SQuAD 2.0 dataset across various k settings and the models. According to appendix A, in SQuAD 2.0 with `gpt-4o-mini`, the highest performance is achieved with the no_k setting, followed by the $k = 5$ setting. From the distribution table, we can observe that the evidence distribution for the no_k setting closely resembles that of the $k = 5$ setting. This suggests that when the LLM extracts evidence without being assigned a specific k value, it inherently selects the k value that it considers optimal for performance. In SQuAD 2.0 with `DeepSeek-V3`, according to Appendix A, it can be observed that all settings with a non-zero value of k yield similar performance, whereas the no_k setting (without an explicit restriction on the number of evidences) achieves the highest performance. According to the evidence distribution, those that k is set to a specific number exhibit highly similar evidence distributions, while the no_k setting shows a distinct distribution characterized by a significantly smaller number of evidences. This suggests that when a restriction k is imposed, the model tends to generate approximately k pieces of evidence regardless of whether they are necessary. In contrast, when no such constraint is enforced, the model naturally selects fewer but more relevant evidences, leading to better overall performance.



(a) `gpt-4o-mini` (b) `DeepSeek-V3`

Figure 5: **Comparison of evidence count distribution in the SQuAD 2.0 dataset** for `gpt-4o-mini` and `DeepSeek-V3`

5.4 Computational Cost

We report the computational cost of this framework in terms of LLM usage, which consists of two distinct phases: retrieval and generation, each requiring a single LLM call per query.

Retrieval phase. The retrieval phase requires one LLM call with an input of length $O(n)$, where n is the number of context tokens in the full context. The output is a Python list of retrieved indices of length $O(k)$, where $k \ll n$, representing the top-k relevant sentences.

Generation phase. The generation phase also requires one LLM call, taking the concatenated contents of the k retrieved chunks (whose total length is $O(k')$, where k' is the sum of tokens in the selected contexts) along with the query. Overall, the framework requires two LLM calls per query: one with context of length $O(n)$ and one with retrieved indices of length $O(k')$.

The experiments were conducted on three datasets: LoCoMoQA, QASPER, and SQuAD. The total number of runs for this framework is the number of QA pairs multiplied by the number of top-k settings, resulting in $4{,}585 \times 5 = 22{,}925$ runs. When `DeepSeek-V3` is used as the retriever, it costed around $\$38.3$. This means that each single run requires around $\$0.0016$.

6 Conclusion

In this study, we propose a new method called ***Letriever***. We leveraged Large Language Models (LLMs) as contextual retrievers and explored its potential for Question Answering (QA) tasks. Our findings demonstrate that LLMs can effectively retrieve relevant information and adaptively process long contexts, resulting in higher answer prediction accuracy and retrieval precision compared to embedding-based retrieval methods. This robustness stems from the flexibility of varying top-k settings, which allows the model to retrieve an appropriate number of contexts based on different questions. There are also an ability to include contexts that do not contain exact same words from the question but are logically essential.

Such adaptability enables LLM-based retrieval to address complex contextual nuances more effectively than traditional Retrieval-Augmented Generation (RAG) approaches that retrieve contexts based on embedding similarity.

7 Limitations

Letriever employs a simple two-stage architecture consisting of a retriever and a generator because the paper aims to investigate the potential of LLMs to be used directly as retrievers. However, as a result, inference involves two sequential steps, which increases both latency and computational cost. This characteristic poses challenges for real-world applications that require real-time responses or operate under limited computational resources. Improving the efficiency of this process remains an important direction for future work.

Additionally, as discussed, LLMs struggle to effectively utilize long contexts, which could influence the performance of the retrieval stage in our approach, as the LLM needs to retrieve relevant contexts from the full context. A framework that reduces the number of contexts given in the retrieval stage may be necessary for higher performance. Therefore, the future work could focus on enhancing the retrieval performance of LLMs by reducing the length of the given contexts during the retrieval stage.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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*, volume 33, pages 1877–1901. Curran Associates, Inc.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. 2020. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*.

Zihang Dai. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

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. *arXiv preprint arXiv:2105.03011*.

Soham De, Samuel L Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. 2024. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. *arXiv preprint arXiv:2207.06300*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

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.

Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. 2023. Llatrieval: Llm-verified retrieval for verifiable generation. *arXiv preprint arXiv:2311.07838*.

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: Industry Track*, pages 881–893.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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. *arXiv preprint arXiv:2302.07452*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. *arXiv preprint arXiv:2109.05052*.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070.

Irina Radeva, Ivan Popchev, and Miroslava Dimitrova. 2024. Similarity thresholds in retrieval-augmented generation. In *2024 IEEE 12th International Conference on Intelligent Systems (IS)*, pages 1–7. IEEE.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Teven Le Scao and Alexander M Rush. 2021. How many data points is a prompt worth? *arXiv preprint arXiv:2103.08493*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023a. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023b. Self-knowledge guided retrieval augmentation for large language models. *arXiv preprint arXiv:2310.05002*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Feifan Wu, Lingyuan Liu, Wentao He, Ziqi Liu, Zhiqiang Zhang, Haofen Wang, and Meng Wang. 2024. Time-sensitve retrieval-augmented generation for question answering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2544–2553.

Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing conversational search: Large language model-aided informative query rewriting. *arXiv preprint arXiv:2310.09716*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

A Detailed Experimental Results

The detailed question answering evaluation results with all top-k settings are presented in Table 5, Table 6 and Table 7.

Answer Prediction. The Full Context method showed the worst performance on LoCoMoQA and SQuAD, while in QASPER, all the RAG methods, except for Letriever (Ours), deteriorated compared to Full Context. This indicates limitations of RAG on certain datasets. Regarding Letrievers with different base models, they showed different trends from the perspective of the top_k setting. However, no_k setting consistently showed decent performance across various datasets.

Retrieval Accuracy. Regarding Retrieval Accuracy, Letriever achieved decent performance when k is set from no_k to 10 in both QASPER and LoCoMoQA. Otherwise, when k increases to 25 or 50, embedding-based retrievers generally achieved higher Recall scores than Letriever. The rational for the result is discussed in Experiments 4.4. However, as discussed, a high Retrieval Recall does not necessarily correlate with better Answer Prediction performance. While it is important to include sufficient relevant content, it is equally crucial to reduce noise in the retrieval process to improve the Precision score. Regarding this, Letriever demonstrates its potential by achieving outstanding Precison and F1 scores, escpecailly when $k = no\_k$.

Retrieval Method	Answer Prediction	
	F1	**EM**
Full Context	63.0	31.7
no_k		
Letriever (gpt-4o-mini)	77.4	44.3
Letriever (DeepSeek-V3)	**82.7**	**51.8**
k = 5		
DRAGON	75.2	43.4
$E5_{\text{mistral-7b}}$	74.2	44.3
openai-embedding	74.9	41.7
Letriever (gpt-4o-mini)	75.5	44.6
Letriever (DeepSeek-V3)	79.6	48.5
k = 10		
DRAGON	75.4	44.6
$E5_{\text{mistral-7b}}$	72.1	42.9
openai-embedding	72.3	41.1
Letriever (gpt-4o-mini)	75.1	43.1
Letriever (DeepSeek-V3)	79.6	48.8
k = 25		
DRAGON	72.4	42.6
$E5_{\text{mistral-7b}}$	70.9	41.4
openai-embedding	70.2	39.4
Letriever (gpt-4o-mini)	74.6	43.4
Letriever (DeepSeek-V3)	79.1	48.2
k = 50		
DRAGON	70.5	40.9
$E5_{\text{mistral-7b}}$	71.0	41.1
openai-embedding	69.3	38.6
Letriever (gpt-4o-mini)	74.5	43.1
Letriever (DeepSeek-V3)	78.8	47.7

Table 5: **Detailed question answering performance on SQuAD 2.0 dataset**. The optimal performance is marked in **bold**. Results are based on F1-score, EM metric for answer prediction; higher is better.

Retrieval Method	Answer Prediction				Evidence Retrieval					
	F1		ROUGE-L		Recall		Precision		F1	
	Dia.	Obs.	Dia.	Obs.	Dia.	Obs.	Dia.	Obs.	Dia.	Obs.
Full Context	39.1	28.7	38.5	27.7	**100.0**	**100.0**	0.3	0.5	0.6	1.0
no_k										
Letriever (gpt-4o-mini)	48.7	42.9	48.0	41.9	68.2	56.9	46.1	41.4	55.0	47.9
Letriever (DeepSeek-V3)	51.8	43.7	51.0	42.8	72.6	60.7	52.7	45.1	61.1	51.7
k = 5										
DRAGON	43.2	41.0	42.5	40.0	63.7	61.8	16.1	16.5	25.7	26.0
$E5_{\text{mistral-7b}}$	43.9	39.8	43.1	39.0	62.7	59.7	15.5	15.6	24.9	24.6
openai-embedding	46.4	41.8	45.7	40.9	68.7	63.2	17.5	17.0	27.9	26.8
Letriever (gpt-4o-mini)	48.8	42.8	47.9	41.9	66.6	59.8	33.4	27.8	44.5	38.0
Letriever (DeepSeek-V3)	51.0	**44.4**	50.2	**43.5**	71.0	58.9	**55.3**	**46.0**	**62.2**	51.7
k = 10										
DRAGON	44.4	40.7	43.6	39.8	72.7	66.4	9.6	9.3	17.0	16.3
$E5_{\text{mistral-7b}}$	44.8	40.7	43.9	39.7	71.5	65.8	9.2	9.1	16.3	16.0
openai-embedding	47.3	41.6	46.6	40.7	77.6	68.8	10.3	9.8	18.2	17.2
Letriever (gpt-4o-mini)	48.4	42.7	47.7	41.8	68.0	61.4	31.7	29.0	43.2	39.4
Letriever (DeepSeek-V3)	52.0	44.1	51.1	43.3	72.3	61.9	50.2	45.4	59.3	**52.4**
k = 25										
DRAGON	45.1	39.2	44.2	38.2	81.8	71.9	4.5	4.3	8.5	8.1
$E5_{\text{mistral-7b}}$	44.8	40.4	43.9	39.4	81.3	72.0	4.4	4.3	8.3	8.1
openai-embedding	46.4	41.4	45.7	40.0	87.1	73.3	4.9	4.5	9.3	8.5
Letriever (gpt-4o-mini)	47.1	42.5	46.4	41.6	68.3	63.1	27.7	28.7	39.4	39.5
Letriever (DeepSeek-V3)	**52.2**	43.2	**51.3**	42.3	75.7	62.5	43.0	42.1	54.8	50.3
k = 50										
DRAGON	44.7	38.5	43.7	37.5	87.5	74.7	2.5	2.3	4.9	4.5
$E5_{\text{mistral-7b}}$	45.2	39.1	44.3	38.1	87.1	75.3	2.4	2.3	4.7	4.5
openai-embedding	46.3	39.6	45.4	38.6	91.9	76.3	2.7	2.4	5.2	4.7
Letriever (gpt-4o-mini)	47.6	42.4	46.7	41.4	73.3	63.5	28.7	30.1	41.2	40.8
Letriever (DeepSeek-V3)	50.2	42.4	49.3	41.6	76.1	62.6	38.6	40.7	51.2	49.3

Table 6: **Detailed question answering performance on LoCoMoQA**. The best performance is marked in **bold**. Results are based on F1-score, ROUGE-L metric for answer prediction and Recall, Precision, F1 metric for evidence retrieval; higher is better.

Retrieval Method	Answer Prediction		Evidence Retrieval		
	F1	**ROUGE-L**	**Recall**	**Precision**	**F1**
Full Context	47.9	46.5	**100.0**	5.3	9.7
			no_k		
Letriever (gpt-4o-mini)	47.9	46.1	52.8	57.9	47.6
Letriever (DeepSeek-V3)	**50.5**	**48.9**	63.5	63.1	52.2
			k = 5		
DRAGON	30.0	29.3	39.1	44.5	35.5
$E5_{\text{mistral-7b}}$	37.1	35.8	48.2	44.0	39.8
openai-embedding	37.2	36.0	43.0	49.2	39.0
Letriever (gpt-4o-mini)	46.6	44.7	51.9	52.9	46.0
Letriever (DeepSeek-V3)	49.7	47.9	56.4	**64.1**	52.7
			k = 10		
DRAGON	36.4	35.2	56.8	35.0	37.5
$E5_{\text{mistral-7b}}$	40.5	38.8	65.0	33.1	38.5
openai-embedding	40.6	38.9	60.9	37.9	40.4
Letriever (gpt-4o-mini)	48.4	46.4	67.1	42.8	45.8
Letriever (DeepSeek-V3)	50.3	48.6	75.4	49.6	**52.9**
			k = 25		
DRAGON	40.8	39.4	76.8	21.6	30.5
$E5_{\text{mistral-7b}}$	44.3	42.2	82.7	19.4	28.7
openai-embedding	43.3	41.8	80.9	23.1	32.5
Letriever (gpt-4o-mini)	48.6	46.8	76.7	35.6	41.2
Letriever (DeepSeek-V3)	48.4	46.9	88.2	27.5	37.9
			k = 50		
DRAGON	42.9	41.2	88.2	13.6	21.9
$E5_{\text{mistral-7b}}$	45.2	43.1	92.0	12.0	19.8
openai-embedding	43.8	42.0	91.1	14.0	22.6
Letriever (gpt-4o-mini)	48.8	47.1	76.6	35.4	39.1
Letriever (DeepSeek-V3)	46.7	44.9	89.2	22.6	30.4

Table 7: **Detailed question answering performance on QASPER**. The best performance is marked in **bold**. Results are based on F1-score, ROUGE-L metric for answer prediction and Recall, Precision, F1 metric for evidence retrieval; higher is better.

B Case Study

B.1 Comparison between Embedding-based Retrieval and Our Method

Question: "What does Jon plan to do at the grand opening of his dance studio?"
Gold Answer: "savor all the good vibes"

Retrieval Method: DRAGON (top-k: 50)

Retrieved Context List:
Top-1. 8:29 pm on 13 June, 2023 - Jon said, "Thanks, Gina! It's been so inspiring to work with our young dancers, seeing their passion and commitment. Opening the dance studio's been a great experience - I want it to be a place of support and encouragement for all our dancers. Will you show me this presentation?"
...
Top-50. 9:32 am on 8 February, 2023 - Jon said, "Thanks, Gina! Your pep-talk really meant a lot. I'm not gonna give up on my dreams - my dance studio and biz ventures need the hard work I'm putting in. Love having you in my corner, thanks for always being there!"

Recall@k: 0.0
Predicted Answer: "Let's make some awesome memories tomorrow at the grand opening!"
F1: 0.0

Retrieval Method: $E5_{\text{mistral-7b}}$ (top-k: 50)

Retrieved Context List:
Top-1. 10:04 am on 19 June, 2023 - Jon said, "Thanks, Gina. Still working on opening a dance studio."
...
Top-50. 3:14 pm on 11 May, 2023 - Gina said, "It must be scary stepping into the unknown but I know you can do it, Jon. With your determination and drive, your dance studio will be a huge success. Keep that positive outlook and keep going!"

Recall@k: 0.0
Predicted Answer: "Let's make some awesome memories tomorrow at the grand opening!"
F1: 0.0

Retrieval Method: openai-embedding (top-k: 50)

Retrieved Context List:
Top-1. 4:04 pm on 20 January, 2023 - Jon said, "I've been into dancing since I was a kid and it's been my passion and escape. I wanna start a dance studio so I can teach others the joy that dancing brings me."
...
Top-50. 10:04 am on 19 June, 2023 - Jon said, "Yeah! Let's make some awesome memories tomorrow at the grand opening!" and shared a photo of a man in a native costume is giving another man a high five

Recall@k: 0.0
Predicted Answer: "Can't wait for it - and for you to be there!"
F1: 0.0

Table 8: **A case study of embedding-based retrieval methods on LoCoMoQA.** All of the methods tended to prioritize contexts that include words in the given question, highlighted in red, but without a logical reason. The methods retrieved 50 contexts, but they failed to retrieve evidence that answers the question.

Question: "What does Jon plan to do at the grand opening of his dance studio?"
Gold Answer: "savor all the good vibes"

Retrieval Method: Letriever (Ours) (top-k: no_k)

Retrieved Context List:

Top-1. 10:04 am on 19 June, 2023 - Jon said, "The official opening night is tomorrow. I'm working hard to make everything just right. Can't wait to see it all come together!" and shared a photo of a group of young dancers in a dance studio

Top-2. 10:04 am on 19 June, 2023 - Gina said, "Congrats, Jon! The studio looks amazing. You've put a lot of work into this and I'm so pumped for the launch tomorrow. Don't miss a beat!"

Top-3. 10:04 am on 19 June, 2023 - Gina said, "Wow, Jon, you must be so excited! You've come so far since we last talked, and tomorrow's gonna be a blast! All those long nights were worth it - so take some time to savor it. Capture the joy and thrill that dance brings - it's magical!"

Top-4. 10:04 am on 19 June, 2023 - Jon said, "Tomorrow's gonna be an awesome night and I'm not gonna forget a second of it. I put so much into this and I want to savor all the good vibes. Thanks for always having my back. You're the best!"

Top-5. 10:04 am on 19 June, 2023 - Jon said, "Yeah! Let's make some awesome memories tomorrow at the grand opening!" and shared a photo of a man in a native costume is giving another man a high five

Recall@k: 100.0
Predicted Answer: "Savor all the good vibes."
F1: 100.0

Table 9: **A case study of the LLM-based retrieval method (Ours) on LoCoMoQA.** LLM was able to retrieve evidence that does not include words in the given question, highlighted in **bold**, but logically supports it. In this case, we did not specify the number of contexts to retrieve, and LLM retrieved 5 contexts, including that the grand opening is tomorrow and Jon plans to savor all the good vibes tomorrow.

B.2 Comparison between Full Context and Our Method

Prompt

```
{context}

Based on the above context, write an answer in the form of a short
phrase for the following question. Answer with exact words from the context
whenever possible. If the answer is date, please specify dates precisely rather
than using relative terms like 'next month'.

Question: {question} Short answer:
```

Table 10: **Prompt used in the generation stage of this case study.** The prompt instructs to provide a concise answer. Therefore, the prediction intended to be concise regardless of the length of the context.

Example 1

Question: "What was the Norman religion?"
Gold Answer: "Catholicism"

Retrieval Method: Full Context
Predicted Answer: "The Normans replaced the Norse religion with Catholicism (Christianity)."
F1: 25.0

Retrieval Method: Letriever (Ours) (top-k: 5)

> Retrieved Context List:
> ...
> Top-2. Rollo's contingents who raided and ultimately settled Normandy and parts of the Atlantic coast included Danes, Norwegians, Norse–Gaels, Orkney Vikings, possibly Swedes, and Anglo-Danes from the English Danelaw under Norse control.The descendants of Rollo's Vikings and their Frankish wives would replace the Norse religion and Old Norse language with Catholicism (Christianity) and the Gallo-Romance language of the local people, blending their maternal Frankish heritage with Old Norse traditions and customs to synthesize a unique N̈ormanc̈ulture in the north of France.
> ...

Predicted Answer: "Catholicism (Christianity)"
F1: 66.7

Example 2

Question: "When did ARPNET and SITA become operational?"
Gold Answer: "1969"

Retrieval Method: Full Context
Predicted Answer: "ARPANET and SITA became operational in 1969."
F1: 28.6

Retrieval Method: Letriever (Ours) (top-k: 5)

> Retrieved Context List:
> Top-1. <Uyless Black, X.25 and Related Protocols, IEEE Computer Society, 1991> <Uyless Black, Frame Relay Networks, McGraw-Hill, 1998> <Uyless Black, MPLS and Label Switching Networks, Prentice Hall, 2001> < Uyless Black, ATM, Volume I, Prentice Hall, 1995>ARPANET and SITA HLN became operational in 1969.

Predicted Answer: "1969"
F1: 100.0

Table 11: **Case Study of the LLM-based retrieval method (Ours) on SQuAD 2.0.** When the full context is provided, the predictions tend to include the correct answer but are delivered in long sentences rather than concise responses, which are intended by the prompt. On the other hand, the Letriever (Ours) answers succinctly and provides the correct answer effectively.

C Assessment of Contexts Retrieved by LLM

We provide an analysis on the number of contexts retrieved by LLM in Table 12. Overall, the average number of retrieved contexts increases as the top-k increases. However, the increase in the number of retrieved contexts is not proportional to the increase in k. In particular, for the Locomo and SQuAD datasets, even when k is set to 50, the average number of retrieved contexts does not increase significantly. This suggests that when using an LLM as a retriever, the number of retrieved contexts is implicitly regulated by the model's reasoning ability. In other words, the LLM tends to retrieve only the contexts that are essential for answering the given question. In contrast, the QASPER dataset shows a different trend. Since its questions typically require a relatively large amount of context to answer, both `gpt-4o-mini` and `DeepSeek-V3` tend to retrieve a number of contexts that increases nearly proportionally with the given k.

		gpt-4o-mini		**DeepSeek-V3**	
Dataset	**given k**	**Median**	**Average**	**Median**	**Average**
LoCoMoQA (Dialog)	5	5	3.7	1	2.3
	10	4	5.3	2	3.4
	25	5	9.1	3	7.0
	50	4	11.4	3	11.1
	no_k	2	28.7	2	9.0
LoCoMoQA (Observation)	5	5	4.0	1	2.2
	10	5	5.3	1	3.1
	25	4	7.5	2	5.5
	50	4	8.3	2	7.6
	no_k	2	3.4	1	5.4
QASPER	5	5	5.0	5	5.7
	10	10	9.3	10	9.7
	25	25	19.0	25	24.5
	50	10	35.6	50	42.9
	no_k	5	5.6	5	13.4
SQuAD	5	5	4.6	1	2.4
	10	10	7.3	1	3.2
	25	5	11.4	1	4.5
	50	5	11.8	2	6.0
	no_k	2	2.8	1	3.9

Table 12: **Comparison of the actual number of retrieved evidences under different *k* settings.** For each dataset (LoCoMo, QASPER, SQuAD), we report the median and average number of evidences actually retrieved by Letriever under different values of k or in the no-k setting.