AttentionRetriever: Attention Layers are Secretly Long Document Retrievers

Anonymous ACL submission

Abstract

Retrieval augmented generation (RAG) has been widely adopted to help Large Language Models (LLMs) to process tasks involving long documents. However, existing retrieval models are not designed for long document retrieval and fail to address several key challenges of long document retrieval, including context-awareness, causal dependence, and scope of retrieval. In this paper, we proposed AttentionRetriever, a novel long document retrieval model that leverages attention mechanism and entity-based retrieval to build context-aware embeddings for long document and determine the scope of retrieval. With extensive experiments, we found AttentionRetriever is able to outperform existing retrieval models on long document retrieval datasets by a large margin while remaining as efficient as dense retrieval models.

1 Introduction

Recent advancements in Large Language Models (LLMs) (OpenAI, 2023, 2025; Dubey et al., 2024; Jiang et al., 2023) have demonstrated strong capabilities in natural language understanding, but recent studies (Liu et al., 2024; Maharana et al., 2024; Lu et al., 2024) have shown that LLMs still struggle to perform well on long document processing tasks due to the lost-in-the-middle problem and limited context window length, while the quadratic complexity of the attention mechanism makes processing long documents very expensive, especially for large models with hundreds of billions of parameters. In recent years, retrieval-augmented generation (RAG) (Lewis et al., 2020) techniques have been widely applied to address this issue by selecting relevant information from the document with a retrieval model, which improves performance by removing distracting information and decreases processing time by shortening the input length.

In the RAG pipeline, the retrieval model needs to perform **long document retrieval**, which requires the model to find a subset D' from a user-provided long document D such that $|D'| \ll |D|$ and D' is sufficient and necessary to answer the input query q. However, existing retrieval models are not tailored for long document retrieval and overlook the following three types of dependencies in long documents:

- **Contextual dependency.** Since long documents are generally coherent, context is often required to resolve issues like coreference and word ambiguity, which are crucial for determining the relevance of chunks. For example, in a document discussing Chicago, the author might use "the city" to refer to "Chicago", but this reference is clear only if the context is provided.

- **Causal dependency.** The query may involve intermediate answers from the document that are needed to reach the final answer. For the same document about Chicago, an example query would be "What was the population of Chicago when the Great Fire happened?", where the intermediate answer "the Great Fire happened in 1871" is needed to find the chunk containing the final answer.

- **Query dependency.** Text chunks providing background information, like the one containing "the Great Fire happened in 1871" in the previous example, are also important to answering the query and should be retrieved. However, these chunks might receive low similarity scores because they are not very relevant to the query, which asks about "the population of Chicago". Therefore, it is necessary to accurately decide the scope of retrieval in long document retrieval tasks.

Modeling the first two dependencies requires a more advanced retrieval model that can build

context-aware representations and update the embeddings as additional contextual information is available. We found that the attention layers in transformer models perfectly match both requirements. Since attention layers calculate the representations of each token by aggregating information from other tokens, they are essentially cross-encoders that embed contextual information into the representation of each token, providing more abundant semantic information compared to existing embedding-based retrieval models. Furthermore, as the representations are propagated through layers, they are also dynamically adjusted based on contextual information gathered in previous layers to encode causal dependencies.

It is also intuitive to use attention layers as retrievers because the attention operations are essentially calculating similarity scores. In each attention layer of the transformer model, the attention score assigned to the j-th token by the i-th token is calculated as the weighted dot product between the key vector k_j and the query vector q_i, which is identical to similarity calculation of embedding models. Moreover, attention computation is performed on two sets of embeddings q and k, which allows the attention layers to perform a broader range of tasks other than semantic similarity search by adjusting the embeddings.

However, training a transformer model for retrieval is very expensive. Ye et al. (2025) found that the last layer in the Qwen-2 model shows high retrieval accuracy without any additional training, implying the possibility of directly employing pretrained LLMs to estimate relevance with attention scores. However, their experiments were very limited and the findings might not generalize to other attention layers and other LLMs. More importantly, since pretrained LLMs suffer from accuracy and efficiency issues with long context, it is also crucial to find out whether attention scores can be efficiently calculated and still achieve high retrieval accuracy with long context.

Therefore, we conducted careful analysis (Section 3) to verify if attention layers in pretrained LLMs can be effective and efficient training-free retrievers. Our analysis showed that **attention scores are more precise than outputs in collecting relevant information and suffer less from the lost-in-the-middle problem**, validating the effectiveness of using LLMs for retrieval tasks despite that they face various issues in long context tasks. Moreover, existing attention approximation methods for LLMs can be directly applied to this approach, making it possible to process long documents with arbitrary lengths. Furthermore, we found that employing pretrained LLMs with around 3 billion parameters as retrievers is already able to achieve impressive performance, which eliminates the need for using larger LLMs and improves efficiency.

However, attention scoring alone is still insufficient to model the third dependency. Background information is still unlikely to receive high attention scores in layers with high retrieval accuracies because it is already embedded into the representations before these layers. To obtain a better estimate of the scope of retrieval, it is essential to additionally consider text chunks that are not immediately relevant but provide background information. Since each piece of background information typically focuses on one entity, and it should be included as part of the retrieval result only if the entity is relevant to the input query, we believe an entity graph structure could help to determine the retrieval scope precisely by connecting text chunks through entities and finding the entities relevant to the query during retrieval to discover hidden background information. In contrast to knowledge graphs, entity graphs are much easier and more efficient to construct, without the need to extract relationships between entities.

Based on these findings and analysis, we proposed AttentionRetriever, a novel retrieval model that builds context-aware embeddings with pretrained LLMs and decides the scope of retrieval through entity-based retrieval, as summarized in Figure 1. During retrieval, we leverage LLMs to process the long document and the query together, adopting the attention maps at layers that show high retrieval accuracies to estimate the relevance score of each text segment, which is combined with embedding-based similarity scoring for more precise retrieval. To decide the scope of retrieval, we find the desired entities by ranking them by the scores of sentences containing the entities. We eventually obtain the final output by collecting all text chunks that contain the highest ranked entities and sentences.

To better evaluate and compare the performance and efficiency of our proposed method and baselines on extremely long documents, we also constructed a new long document retrieval dataset consisting of different types of documents with an average length of over 100,000 words and various types of queries (Section 5). To the best of our

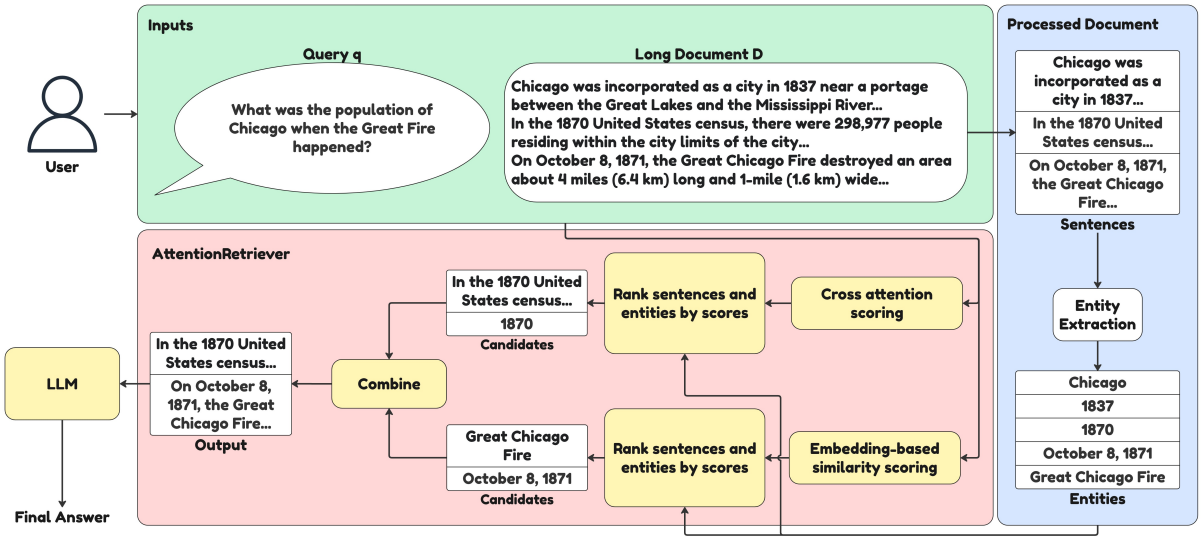


Figure 1: Overview of AttentionRetriever.

knowledge, this is the first retrieval dataset that features documents with lengths exceeding the context window length of most existing LLMs.

We evaluated AttentionRetriever on our constructed dataset, as well as six other single-document retrieval datasets and three multi-document retrieval datasets. AttentionRetriever outperforms state-of-the-art sparse and dense retrieval models by a large margin on single-document retrieval datasets while remaining as efficient as dense retrieval models with similar sizes. Moreover, AttentionRetriever also achieves competitive performance in multi-document retrieval datasets where contextual information is generally not needed, further demonstrating its effectiveness in long document retrieval tasks.

In summary, our contributions are as follows:

- We propose AttentionRetriever, which leverages the attention mechanism and entity-based retrieval to perform context-aware long document retrieval and decide the scope of retrieval based on the query;
- We perform empirical analysis on the attention mechanism in pretrained LLMs, verifying the effectiveness of using attention maps for retrieval tasks and its capability of dynamically updating embeddings across layers;
- We construct a new long document retrieval dataset consisting of extremely long documents to compare the retrieval accuracy of our proposed method and various baselines.

2 Related Works

2.1 Long Document Retrieval

Although the concept of text retrieval has appeared for many years, the task of long document retrieval remains largely unexplored. Existing sparse and dense models, such as BM25 (Robertson and Zaragoza, 2009), DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2021), GTR (Ni et al., 2022), mGTE (Zhang et al., 2024), and Grit-LM (Muennighoff et al., 2025), are mostly designed for open-domain retrieval, in which the models deal with a large corpus of independent documents instead of a long document and can process each document separately because other ones are likely to be irrelevant. Researchers have attempted to incorporate context-awareness into retrieval models (Morris and Rush, 2024; Günther et al., 2024; Conti et al., 2025), but they are still designed for open-domain retrieval. In recent years, SPScanner (Cao et al., 2025) and MC-Indexing (Dong et al., 2024) have been proposed to address the problem of long document retrieval. However, they still failed to address the challenges of causal and query dependencies.

2.2 Context Window Length Extension

LLMs fail to process inputs exceeding its context window length because they lack training on out-of-distribution (OOD) indices. To tackle this problem, prior works (Ding et al., 2024; Shang et al., 2025; An et al., 2024; Jin et al., 2024; Xu et al., 2025; Liu et al., 2025) proposed to map OOD indices to in-distribution indices to avoid the problem. To reduce the computational cost when processing

long contexts, researchers also proposed methods to approximate the full attention map by dividing the context into text segments and performing attention operations only on relevant text segments (Xiao et al., 2024; Fountas et al., 2024; Willette et al., 2025).

2.3 Attention Mechanism Interpretation

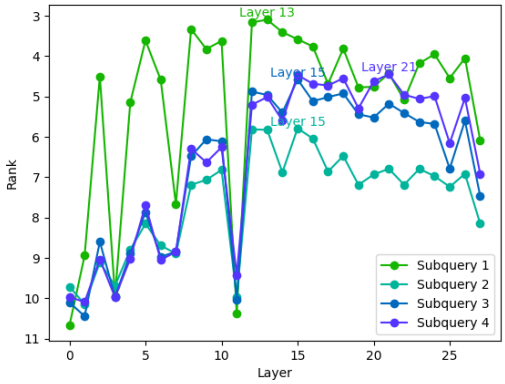
Despite the success of modern LLMs, it remains unclear how the attention mechanism (Bahdanau et al., 2015) in LLMs contribute to its success. Vig and Belinkov (2019) showed that different attention heads are assigned to different tasks in GPT-2, while Sun et al. (2024) also found that the middle transformer layers have similar functionalities and are responsible for different tasks. Ye et al. (2025) conducted analysis on the final layer of Qwen-2 model and concluded that high attention scores are given to tokens relevant to the query. However, no existing work has explored the potential of employing attention layers in LLMs for retrieval.

3 Observations

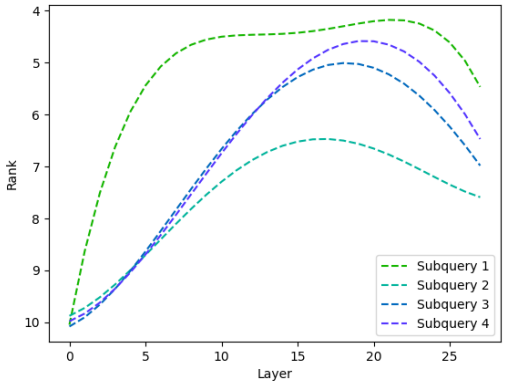
In order to verify whether the attention layers of different LLMs can accurately find relevant chunks for different types of queries, we conducted a detailed empirical analysis to measure the effectiveness of each attention layer for retrieval and the shift of attention patterns across layers. We selected LLaMA-3.2 3B (Dubey et al., 2024), Qwen-2.5 3B (Yang et al., 2024), and Mistral 7B (Jiang et al., 2023) as representatives of LLMs, and analyzed the cross attention between the query and the document at each attention layer when the document and the query are processed by the LLMs. To compare the attention patterns of different types of queries, we utilized the training set of MuSiQue dataset (Trivedi et al., 2022), which contains different types of multi-hop questions and can be decomposed into simpler subqueries.

In each attention layer, we calculate its retrieval accuracy by finding the paragraphs the model focuses on through cross attention scores. Specifically, we estimate the relevance of each paragraph by the maximum cross attention scores assigned by the query to tokens from the paragraph, averaged over all attention heads. Formally, given the attention map $A \in \mathbb{R}^{H \times T_d \times T_q}$, the paragraph score $score_p$ is calculated as:

$$score_p = \max_{s_l \le t \le s_r, 1 \le t_q \le T_q} \left(\frac{1}{H} \sum_{h=1}^{H} A_{h,t,t_q} \right)$$



(a) Average ranks of the gold paragraph for each subquery over all queries in the dataset. The layer achieving the highest average rank for each subquery is marked at the top of each line. The subqueries are ordered by dependencies, where subquery 1 does not depend on any other subqueries while subquery 4 could depend on any other subqueries.

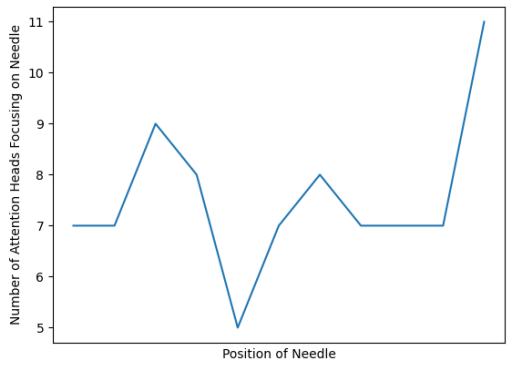


(b) The quartic approximation of the average ranks.

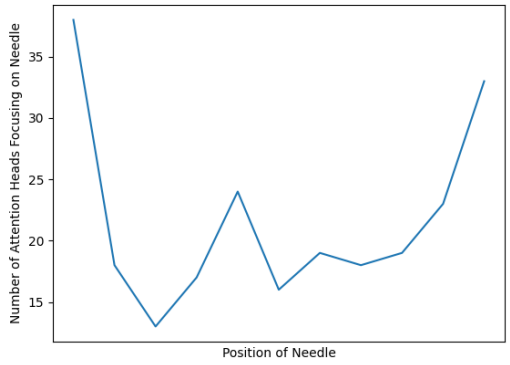
Figure 2: Results of attention analysis.

where H is the number of attention heads, T_d is the length of the document, T_q is the length of the query, and the paragraph p spans over tokens $p_l, \ldots, p_r$. We then rank the paragraphs by their corresponding scores to determine the paragraphs being focused on and compare with the gold paragraphs for each subquery of the questions to calculate the retrieval accuracy for the subquery.

Figure 2 presents the results of attention layers of `Llama-3.2-3B-Instruct` on all queries, while the results for other models and each specific type of queries can be found in Appendix A. The top figure shows the average ranking of the gold paragraph for each subquery across the layers, while the bottom figure is the quartic approximation of the rankings. From Figure 2a, we found that **only certain attention layers can achieve high retrieval accuracy**, and they are mostly in the second half



(a) Retrieval accuracy for attention layers.



(b) Retrieval accuracy for attention layers with Cascading KV Cache.

Figure 3: Test results on the needle-in-a-haystack test.

of the layers. By analyzing the approximation in Figure 2b, we also found that **pretrained LLMs shift focuses in different attention layers**. Earlier attention layers tend to focus on the independent subquery (subquery 1), while later layers rank the gold paragraphs for subqueries 2, 3, and 4 higher, which depends on subquery 1. This result validates the process of building context and causal dependencies through attention layers in LLMs.

However, pretrained LLMs are known to struggle with processing long documents due to lost-in-the-middle problem and limited context window length. Therefore, we conducted additional experiments on the needle-in-a-haystack (gkamradt) test to verify if lost-in-the-middle also appears in attention layers and if existing context length extension methods can be applied to our pipeline. To determine whether the attention layers can find the needles, we calculated the number of attention heads where the highest attention score is assigned to the needle. We selected `Llama-3.2-3B-Instruct` as the base model and employed Cascading KV Cache (Willette et al., 2025) as the context extension method, and tested on documents with approximately 100,000 tokens.

Figure 3 shows the results of our experiments. Figure 3a reveals that the count does not decrease when the needle is in the middle of the document, indicating that **attention layers are less affected by the lost-in-the-middle problem**. Figure 3b also demonstrates that the approximation method can be effective in our proposed pipeline because it can even find needles much more accurately than full attention. Based on these observations, we proposed our attention-based retrieval approach, which will be outlined in the next section.

4 Method

4.1 Overview

Our proposed retrieval model is summarized in Figure 1. We leverage a pretrained LLM to assign a score for each sentence in the document based on attention maps of the LLM (Subsection 4.2). Meanwhile, we also employ a dense embedding model to calculate a separate score for each sentence by similarity between the sentence embedding the and query embedding to further improve retrieval accuracy (Subsection 4.3). We then use these two sets of scores to find relevant entities and perform an entity-based retrieval to obtain the outputs (Subsection 4.4).

4.2 Attention for Sentence Scoring

The observations of Section 3 suggest that the cross-attention scores between the query and the document in pretrained LLMs could provide an accurate estimate of query relevance. Based on this conclusion, we proposed to employ a pretrained LLM to estimate the relevance of each sentence by computing cross-attention scores between tokens corresponding to the query and tokens from the sentence. Since only certain layers achieve high retrieval accuracies, we only use the attention scores from layers that obtained smallest average ranking for at least one subquery (the layers labeled in Figure 2a) to remove the noise of other layers.

We followed the same procedure as Section 3 to convert token-level attention scores into sentence scores to fully utilize our findings. After obtaining the raw attention map $A \in \mathbb{R}^{L \times H \times T_d \times T_q}$, where L is the number of selected layers, H is the number of attention heads, T_d is the length of the docu-

ment, and T_q is the length of the query, we calculate the attention score for each sentence by taking the maximum attention score assigned to tokens in the sentence across all selected layers and all query tokens, averaged over attention heads. More formally, given a sentence s spanning over tokens $s_l, \ldots, s_r$, the attention score of the sentence a_s is computed as follows:

$$a_s = \max_{1 \le l \le L, s_l \le t \le s_r, 1 \le t_q \le T_q} \left(\frac{1}{H} \sum_{h=1}^{H} A_{l,h,t,t_q} \right)$$

In Section 3, we also found that Cascading KV cache method (Willette et al., 2025) can be seamlessly applied to our method, so we apply this method to handle inputs of that could exceed the pre-defined context window limit of our selected base model efficiently. In practice, since the extension method would bring additional overhead to the pipeline, we only apply it when the length of the document exceeds the context window limit of the base LLM.

4.3 Sentence Embedding for Multi-view Similarity Search

Attention-based similarity estimation provides a token-level measurement of the relevance to the query, which is in fact complementary to the traditional embedding-based sentence-level estimation. To enhance our retrieval process, we use sentence embedding to obtain a sentence-level similarity estimate as an additional view in similarity search to enrich the information obtained by attention-based search. We encode each sentence s into its embedding $E_s = f(s)$ through an embedding model f, and use the same embedding model to obtain the query embedding query embedding $E_q = f(q)$. The sentence embedding score e_s for each sentence s is calculated as the cosine similarity between its embedding E_s and query embedding E_q:

$$e_s = \frac{E_s \cdot E_q}{||E_s||||E_q||}$$

4.4 Entity-based Retrieval

In long document retrieval, it is insufficient to only retrieve the most relevant text chunks because long documents are typically coherent and other chunks could provide additional background information about the query. Since chunks mentioning the entities relevant to the query can usually provide useful information, we find the most relevant entities and use them to find these indirectly relevant text chunks.

Specifically, we use SpaCy (Honnibal et al., 2020) to extract the entities in each sentence, and assign a relevance score to each entity based on the scores of the sentences it appears in to find the most relevant entities. Since relevant entities should only appear in relevant sentences, we calculate the relevance score of entities by the average relevance scores of the sentences.

Since we obtained both attention scores and embedding scores and the attention score and embedding score are not directly comparable, we perform retrieval on each score separately and combine the results with equal weights. With a `top_k` value of k, we first retrieve $\lceil \frac{k}{2} \rceil$ entities and sentences with the highest attention scores and $\lfloor \frac{k}{2} \rfloor$ entities and sentences with the highest embedding scores. The union of these two sets of entities and sentences is the collection of all selected entities and sentences. For each selected sentence, we retrieve the paragraph it belongs to, while for each selected entity, we retrieve all paragraphs containing the entity.

5 Dataset Construction

Since the average length of documents in existing retrieval datasets are very limited, we collected a set of long documents from LongBench-v2 dataset (Bai et al., 2025), which features long documents from a variety of sources, to compare our proposed method with baselines on extremely long documents. We only selected documents with length labeled as `"medium"` or `"long"`, and sub-domain is one of `Financial`, `Academic`, `Governmental`, and `Legal` to obtain more well-structured documents. We collected a total of 35 documents from the dataset.

Through our analysis on prior retrieval and QA datasets, such as MuSiQue (Trivedi et al., 2022), Qasper (Dasigi et al., 2021), and LongBench (Bai et al., 2024), we found that most questions in the datasets fall into the following four categories:

- Single-hop: the question can be answered with a single piece of information.
- Comparison: the question requires comparison between two or more pieces of information from multiple paragraphs.
- Composition: the question composes multiple single-hop questions, and the LLM needs

Average Length	QASA 4665.09	Qasper 3442.26	RepLiQA 970.50	ConditionalQA 1298.13	NaturalQuestions 2548.97	LongBench-v2-Retrieval 106025.49	Average
Sparse Models							
BM25	0.3795	0.2304	0.4896	0.1988	0.3055	0.3126	0.3194
Dense Models							
DPR	0.2181	0.0529	0.3602	0.1542	0.2981	0.1226	0.2010
ANCE	0.3724	0.2856	0.4686	0.1893	0.3818	0.3240	0.3370
CDE	0.1341	0.0962	0.2449	0.1547	0.2416	0.0485	0.1532
GTR	0.3977	0.2714	0.4851	0.2584	0.4110	0.3260	0.3583
GTE-Qwen2	0.3785	0.2442	0.4785	0.2415	0.4131	0.3314	0.3479
Qwen3	0.4414	0.2057	0.4846	0.2892	0.4091	0.3205	0.3584
GritLM	0.4394	0.3008	0.5141	0.3258	0.4592	0.3398	0.3965
Autoregressive Models							
SPScanner	0.4604	0.3712	0.6434	0.1354	0.4237	0.4188	0.4088
AttentionRetriever (Ours)							
LLaMA-3.2 3B	0.5584	**0.4618**	0.8339	**0.3526**	**0.5998**	**0.4738**	**0.5467**
Qwen-2.5 3B	**0.5663**	0.4551	**0.8422**	0.3106	0.5655	0.3215	0.5102

Table 1: Comparison of proposed method and baselines on single-document retrieval datasets, where best and second best results are marked in **bold** and underline, respectively.

to answer each sub-question in sequence to arrive at the final answer.

- Summarization: the question requires summarizing information from multiple locations throughout the document.

Therefore, to ensure our dataset includes all these four types of queries, we manually generated a query for each of these four types of queries for each document we collected, and annotated the corresponding answer and relevant paragraphs. We denote our dataset as **LongBench-v2-Retrieval** for the rest of the paper. A comparison of our dataset and other long document retrieval datasets can be found in Appendix B, which shows that the documents in our new dataset are significantly longer than other retrieval datasets and some of them even exceed the context window length of most LLMs (Dubey et al., 2024; Jiang et al., 2023; Zhang et al., 2025).

6 Experiments

6.1 Experimental Setup

To comprehensively evaluate the effectiveness and efficiency of our proposed method, we conducted our experiment on both long document retrieval tasks and long document question answering (QA) tasks with the retrieval-augmented generation (RAG) setting.

Baselines. We compared AttentionRetriever with the following three types of existing retrieval models that can be applied to long document retrieval:

(a) Sparse retrieval model BM25 (Robertson and Zaragoza, 2009);

(b) Dense retrieval models, such as DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2021), GTR (Ni et al., 2022), GTE-Qwen2-7B (Zhang et al., 2024), GritLM-7B (Muennighoff et al., 2025), and Qwen3-7B (Zhang et al., 2025);

(c) Long document retrieval model SPScanner-1.3B (Cao et al., 2025);

For evaluation on QA tasks, we used pretrained LLMs, such as LLaMA-3.1 8B (Dubey et al., 2024), Mistral-7B (Jiang et al., 2023), Qwen-2.5 7B (Yang et al., 2024), and GPT-5 mini (OpenAI, 2025), as our baselines to compare between RAG setting and direct generation.

Datasets. We conducted experiments on five long document retrieval benchmarks: LongBench-v2-Retrieval (Ours), QASA (Lee et al., 2023), Qasper (Dasigi et al., 2021), RepLiQA (Monteiro et al., 2024), and DAPR (Wang et al., 2024). For DAPR, we only selected datasets where each query can be answered by a single document, which are ConditionalQA, MS MARCO, and Natural Questions. However, we decided not to use MS MARCO because the average paragraph count of documents in the datasets is very limited (as shown in Table 3), and is not suitable for our task. We report the performance of our proposed method and the baselines on the test sets of these benchmarks. For LongBench-v2-Retrieval and RepLiQA, we used the full datasets because the dataset splits are not provided.

We also used three multi-document retrieval benchmarks for further comparison, which are HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). We report the performance of our proposed method and the baselines on the validation sets of these benchmarks.

For evaluation on the QA task, we utilized the LongBench (Bai et al., 2024) dataset and selected three single-document QA subsets, Qasper, MultiFieldQA-en, and NarrativeQA.

Metrics. To evaluate and compare retrieval and QA performance, we use **F-1** as the evaluation metric. For comparison of retrieval efficiency, we measure the average processing time for each sample in the datasets. For QA tasks, we report the number of input tokens passed into the QA model, as we cannot measure the processing time for closed-source models.

Implementation Details. In our experiments, we use paragraphs as basic retrieval units. For baselines that cannot process long paragraphs, we split each paragraph into text chunks that can fit into the context window and take the maximum similarity of the text chunks with the query as the similarity of the paragraph. For AttentionRetriever, we used `Llama-3.2-3B-Instruct` and `Qwen/Qwen2.5-3B-Instruct` as our base LLMs to compute attention scores. For `top_k` choices, we used 1, 2, 3, and 5 because the average number of evidences of the datasets we used are below 5 (as shown in Table 3). For QA tasks, we also used vLLM (Kwon et al., 2023) for efficient generation. We performed all our experiments on a single NVIDIA A40 GPU.

6.2 Main Results

Table 1 and 2 present the retrieval performance of baselines and AttentionRetriever on single-document and multi-document retrieval datasets, respectively. For simplicity, we only included the results with `top_k` value of 3, while the results of other `top_k` values can be found in Appendix C. Our proposed method significantly outperforms the baselines across all single-document retrieval datasets, while achieving a similar performance with baselines on multi-document datasets, which is not the primary target of our method. However, we also noticed a significant performance drop of AttentionRetriever-Qwen on LongBench-v2-Retrieval dataset compared to other dataset, which could indicate that Qwen-2.5 model does not work well with the context extension method. Table 5 compares the retrieval efficiency of AttentionRetriever and the baselines. Our proposed method is as efficient as large dense embedding models like GTE, Qwen3, and GritLM, demonstrating the efficiency of AttentionRetriever.

Average Length	HotpotQA 887.05	2WikiMultihopQA 549.58	MuSiQue 8089.59	Average
Sparse Models				
BM25	0.5695	0.5454	0.3412	0.4854
Dense Models				
DPR	0.5020	0.5389	0.3680	0.4696
ANCE	0.5547	0.6603	0.4285	0.5478
CDE	0.3932	0.2637	0.2320	0.2963
GTR	0.6109	0.6393	0.4591	0.5698
GTE-Qwen2	0.6707	<u>0.664</u>	<u>0.5328</u>	<u>0.6225</u>
Qwen3	0.6721	0.6533	0.522	0.6158
GritLM	**0.7096**	**0.6802**	**0.5484**	**0.6461**
Autoregressive Models				
SPScanner	0.6052	0.6268	0.3982	0.5434
AttentionRetriever (Ours)				
LLaMA-3.2 3B	<u>0.7090</u>	0.6495	0.5084	0.6223
Qwen-2.5 3B	0.7037	0.6355	0.5062	0.6151

Table 2: Comparison of proposed method and baselines on multi-document retrieval datasets, where best and second best results are marked in **bold** and <u>underline</u>, respectively.

The results for the QA tasks can be found in Appendix E. Our proposed method achieves comparable performance as direct generation with the LLM while significantly reducing the number of input tokens, while also outperforming the long document retrieval baseline SPScanner. Notably, we found that all RAG methods fail to perform well on NarrativeQA dataset, which might indicate that RAG methods do not work well on novels since novels are typically less well-structured than other types of long documents and retrieval cannot collect all pieces of information required to answer the question.

7 Conclusion

In this paper, we introduced AttentionRetriever, a training-free context-aware retrieval model that leverages the attention mechanism in LLMs and entity-based retrieval for accurate long document retrieval. In contrast to prior sparse and dense retrieval models, our approach considers contextual, causal, and query dependencies in the long document when evaluating the relevance of each text segment, enabling a more precise retrieval process. Our experiment results reveal that AttentionRetriever can achieve very strong performance on various long document retrieval benchmarks, while maintaining competitive performances on multi-document retrieval scenarios.

Limitations

We identified several major limitations of our work. Firstly, our proposed method requires a sufficiently large LLM (around 3 billion parameters) to perform well, making it less efficient than sparse and small dense models. Secondly, due to hardware constraints, we did not extend our analysis on attention to larger LLMs, which could show different attention patterns. Lastly, due to the scarcity of well-formatted long documents and the difficulty of manual labeling, the size of the retrieval dataset we constructed is limited and can be sensitive to outliers, causing the evaluation results to be inaccurate.

References

Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. 2024. Training-Free Long-Context Scaling of Large Language Models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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. 2024. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3119–3137. 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. 2025. LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-context Multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 3639–3664. Association for Computational Linguistics.

Weili Cao, Jianyou Wang, Youze Zheng, Longtian Bao, Qirui Zheng, Taylor Berg-Kirkpatrick, Ramamohan Paturi, and Leon Bergen. 2025. Single-Pass Document Scanning for Question Answering. *CoRR*, abs/2504.03101. ArXiv: 2504.03101.

Max Conti, Manuel Faysse, Gautier Viaud, Antoine Bosselut, Céline Hudelot, and Pierre Colombo. 2025. Context is Gold to find the Gold Passage: Evaluating and Training Contextual Document Embeddings. *CoRR*, abs/2505.24782. ArXiv: 2505.24782.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A Dataset of Information-Seeking Questions and Answers Anchored in Research Papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4599–4610. Association for Computational Linguistics.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. LongRoPE: Extending LLM Context Window Beyond 2 Million Tokens. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Kuicai Dong, Derrick-Goh-Xin Deik, Yi Lee, Hao Zhang, Xiangyang Li, Cong Zhang, and Yong Liu. 2024. MC-indexing: Effective Long Document Retrieval via Multi-view Content-aware Indexing. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 2673–2691. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783. ArXiv: 2407.21783.

Zafeirios Fountas, Martin Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou-Ammar, and Jun Wang. 2024. Human-like Episodic Memory for Infinite Context LLMs. *CoRR*, abs/2407.09450. ArXiv: 2407.09450.

gkamradt. gkamradt/LLMTest_needleinahaystack: Doing simple retrieval from LLM models at various context lengths to measure accuracy.

Michael Günther, Isabelle Mohr, Bo Wang, and Han Xiao. 2024. Late Chunking: Contextual Chunk Embeddings Using Long-Context Embedding Models. *CoRR*, abs/2409.04701. ArXiv: 2409.04701.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *CoRR*, abs/2310.06825. ArXiv: 2310.06825.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. LLM Maybe LongLM: SelfExtend LLM Context Window Without Tuning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-In Lee, and Moontae Lee. 2023. QASA: Advanced Question Answering on Scientific Articles. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19036–19052. PMLR.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Trans. Assoc. Comput. Linguistics*, 12:157–173.

Xiaoran Liu, Ruixiao Li, Zhigeng Liu, Qipeng Guo, Yuerong Song, Kai Lv, Hang Yan, Linlin Li, Qun Liu, and Xipeng Qiu. 2025. ReAttention: Training-Free Infinite Context with Finite Attention Scope. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yi Lu, Jing Nathan Yan, Songlin Yang, Justin T. Chiu, Siyu Ren, Fei Yuan, Wenting Zhao, Zhiyong Wu, and Alexander M. Rush. 2024. A Controlled Study on Long Context Extension and Generalization in LLMs. *CoRR*, abs/2409.12181. ArXiv: 2409.12181.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating Very Long-Term Conversational Memory of LLM Agents. 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 13851–13870. Association for Computational Linguistics.

João Monteiro, Pierre-André Noël, Étienne Marcotte, Sai Rajeswar Mudumba, Valentina Zantedeschi, David Vázquez, Nicolas Chapados, Chris Pal, and Perouz Taslakian. 2024. RepLiQA: A Question-Answering Dataset for Benchmarking LLMs on Unseen Reference Content. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

John X. Morris and Alexander M. Rush. 2024. Contextual Document Embeddings. *CoRR*, abs/2410.02525. ArXiv: 2410.02525.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. Generative Representational Instruction Tuning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large Dual Encoders Are Generalizable Retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9844–9855. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774. ArXiv: 2303.08774.

OpenAI. 2025. GPT-5 System Card. Technical report.

Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Ning Shang, Li Lyna Zhang, Siyuan Wang, Gaokai Zhang, Gilsinia Lopez, Fan Yang, Weizhu Chen, and Mao Yang. 2025. LongRoPE2: Near-Lossless LLM Context Window Scaling. *CoRR*, abs/2502.20082. ArXiv: 2502.20082.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. 2024. Transformer Layers as Painters. *CoRR*, abs/2407.09298. ArXiv: 2407.09298.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. \unicode9835 MuSiQue: Multihop Questions via Single-hop Question Composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the Structure of Attention in a Transformer Language Model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 63–76. Association for Computational Linguistics.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2024. DAPR: A Benchmark on Document-Aware Passage Retrieval. 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 4313–4330. Association for Computational Linguistics.

Jeffrey Willette, Heejun Lee, Youngwan Lee, Myeongjae Jeon, and Sung Ju Hwang. 2025. Training Free Exponential Context Extension via Cascading KV Cache. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Song Han, and Maosong Sun. 2024. InfLLM: Unveiling the Intrinsic Capacity of LLMs for Understanding Extremely Long Sequences with Training-Free Memory. *CoRR*, abs/2402.04617. ArXiv: 2402.04617.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xinhao Xu, Jiaxin Li, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. 2025. Extending LLM Context Window with Adaptive Grouped Positional Encoding: A Training-Free Method. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 573–587. Association for Computational Linguistics.

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, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 Technical Report. *CoRR*, abs/2412.15115. ArXiv: 2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Xiaoju Ye, Zhichun Wang, and Jingyuan Wang. 2025. Infinite Retrieval: Attention Enhanced LLMs in Long-Context Processing. *CoRR*, abs/2502.12962. ArXiv: 2502.12962.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval. 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 1393–1412. Association for Computational Linguistics.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *CoRR*, abs/2506.05176. ArXiv: 2506.05176.

A Detailed Attention Analysis

Figures 4, 6, and 8 show the average ranks of gold paragraphs over all queries, while Figures 5, 7, and 9 plot the corresponding quartic approximations for LLaMA, Qwen, and Mistral models, respectively. We found our conclusions in Section 3 generally holds for all three LLMs we analyzed.

Figures 10, 11, and 12 also present the detailed results for each type of queries in the dataset. The graph attached to the right of each figure explains the dependencies of the type, where $a \rightarrow b$ denotes subquery b depends on the answer to subquery a. These results further validated our claim that LLMs adjust the token embeddings through the layers with additional contextual information, since LLMs tend to focus on subqueries depending on other subqueries in later layers so that necessary information has been collected before the layers.

B Details of LongBench-v2-Retrieval

B.1 Comparison with Existing Long Document Retrieval Datasets

Table 3 compares the lengths and number of evidences of existing long document retrieval datasets

Dataset	Size	Avg Length	Max Length	Avg Paragraph Count	Max Paragraph Count	Avg Number of Evidences
QASA (Lee et al., 2023)	518	4665.09	15796	52.45	186	1.55
Qasper (Dasigi et al., 2021)	1307	3442.26	21043	61.94	328	1.75
RepLiQA (Monteiro et al., 2024)	89770	970.50	1799	22.31	58	1.00
DAPR (ConditionalQA) (Wang et al., 2024)	2517	1298.13	7733	105.04	559	4.08
DAPR (MS MARCO)	98466	1059.75	152515	4.05	43	1.03
DAPR (NaturalQuestions)	7220	2438.86	28871	30.14	535	1.11
LongBench-v2-Retrieval (Ours)	140	106025.49	350135	2460.23	13944	2.59

Table 3: Comparison of long document retrieval retrieval datasets. "Average Length" and "Maximum Length" are measured in number of words. For datasets consisting of multiple splits, we count the total number of samples across all splits.

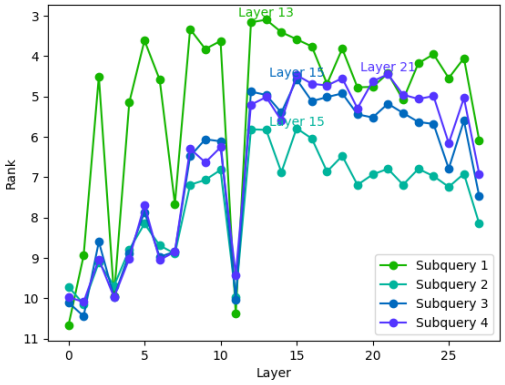


Figure 4: Average ranks of the gold paragraph for each subquery over all queries in the dataset for LLaMA-3.2 3B. The layer achieving the highest average rank for each subquery is marked at the top of each line.

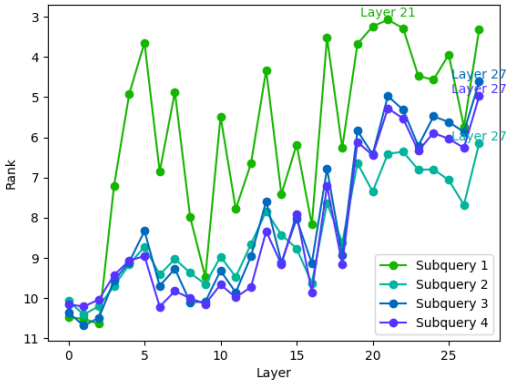


Figure 6: Average ranks of the gold paragraph for each subquery over all queries in the dataset for Qwen-2.5 3B. The layer achieving the highest average rank for each subquery is marked at the top of each line.

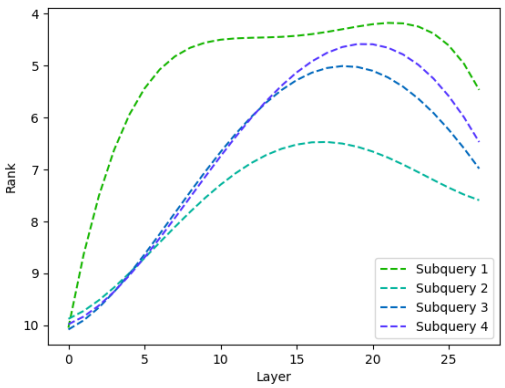


Figure 5: The quartic approximation of the average ranks in Figure 4.

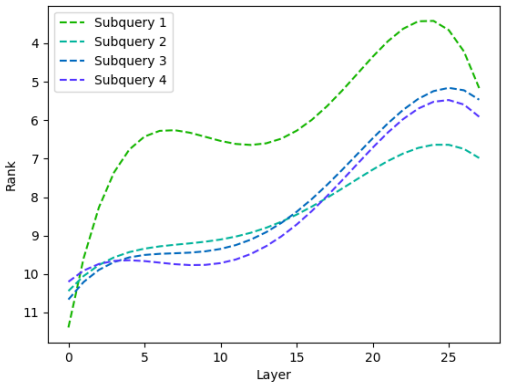


Figure 7: The quartic approximation of the average ranks in Figure 6.

with our newly constructed dataset. The average document length of our dataset is significantly larger than other existing datasets, while the average number of evidences is also larger than most datasets we compared with.

B.2 Example Queries

The following are example queries for each type in our constructed dataset:

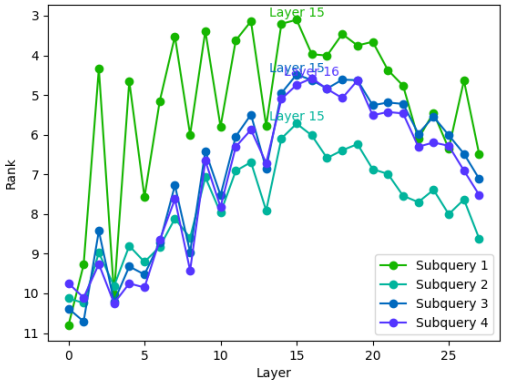


Figure 8: Average ranks of the gold paragraph for each subquery over all queries in the dataset for Mistral 7B. The layer achieving the highest average rank for each subquery is marked at the top of each line.

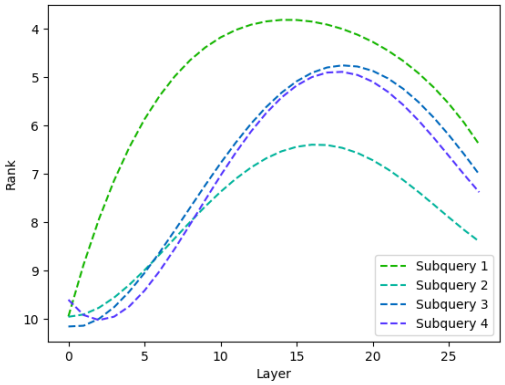


Figure 9: The quartic approximation of the average ranks in Figure 8.

Example Single-hop Query

Question: According to the Report of the Independent High-Level Expert Group on Climate Finance, what is the estimated annual spend on the transformation of the energy system and investing in sustainable agriculture for emerging markets and developing countries by 2030?
Answer: Around $2.4 trillion per year.

Example Comparison Query

Question: Did Czechia receive more funding under EU cohesion policy in 2014-2020 compared to 2007-2013?
Answer: No

Example Composition Query

Question: What are the outstanding health features of the product that had over 2.5 million shipments within three months of its launch?
Answer: Improved Activity Rings and the Stay Fit app to help users build healthier lifestyles.

Example Summarization Query

Question: What are the current challenges of dynamic virtual cluster provisioning in geo-distributed clouds as outlined in the paper?
Answer: There is not yet an efficient resource allocation algorithm for VC provisioning even in the offline case with all user VC requests known, and an efficient pricing mechanism to charge users for the VCs on the go is missing. Moreover, online auction of an entire virtual cluster, including VMs and the network in-between, has not been studied.

C Full Evaluation Results on Retrieval Datasets

The full evaluation results on retrieval datasets with different values of `top_k` can be found in Table 4. Our proposed method is able to consistently outperform the baseline methods across all values of `top_k` we selected.

D Comparison of Efficiency on Single-document Retrieval Datasets

We compare the efficiency of AttentionRetriever with the baselines on long document retrieval datasets in Table 5. Although our method is considerable slower than sparse and small dense models, it is able to achieve similar latency as larger dense models including GTE, Qwen3, and GritLM, especially on datasets with larger average document lengths.

E Evaluation Results on Question Answering Datasets

The results on QA datasets are shown in Table 6. AttentionRetriever achieves comparable performance with direct LLM generation with signifi-

Average Length	QASA 4665.09	Qasper 3442.26	RepLiQA 970.50	ConditionalQA 1298.13	NaturalQuestions 2548.97	LongBench-v2-Retrieval 106025.49	Average
Sparse Models							
BM25							
top_k=1	0.4530	0.1983	0.7862	0.1412	0.3174	0.3576	0.3756
top_k=2	0.4255	0.2260	0.6133	0.1714	0.3284	0.3296	0.3490
top_k=3	0.3795	0.2304	0.4896	0.1988	0.3055	0.3126	0.3194
top_k=5	0.3087	0.2101	0.3475	0.2044	0.2572	0.2703	0.2665
Dense Models							
DPR							
top_k=1	0.1979	0.0326	0.4655	0.1189	0.2743	0.0900	0.1965
top_k=2	0.2138	0.0462	0.4177	0.1470	0.3028	0.1139	0.2069
top_k=3	0.2181	0.0529	0.3602	0.1542	0.2981	0.1226	0.2010
top_k=5	0.2055	0.0608	0.2827	0.1689	0.2694	0.1073	0.1824
ANCE							
top_k=1	0.4216	0.2886	0.7364	0.1804	0.4309	0.3269	0.3975
top_k=2	0.4028	0.2957	0.5826	0.1898	0.4223	0.3500	0.3739
top_k=3	0.3724	0.2856	0.4686	0.1893	0.3818	0.3240	0.3370
top_k=5	0.3019	0.2515	0.3372	0.1945	0.3084	0.2584	0.2753
CDE							
top_k=1	0.1061	0.0810	0.2535	0.1025	0.2117	0.0452	0.1334
top_k=2	0.1199	0.0969	0.2598	0.1419	0.2398	0.0443	0.1504
top_k=3	0.1341	0.0962	0.2449	0.1547	0.2416	0.0485	0.1532
top_k=5	0.1293	0.0987	0.2129	0.1547	0.2211	0.0521	0.1447
GTR							
top_k=1	0.4643	0.2291	0.7568	0.2366	0.4896	0.3118	0.4147
top_k=2	0.4397	0.2695	0.6035	0.2622	0.4632	0.3597	0.3996
top_k=3	0.3977	0.2714	0.4851	0.2584	0.4110	0.3260	0.3583
top_k=5	0.3234	0.2425	0.3466	0.2514	0.3285	0.2846	0.2962
GTE-Qwen2							
top_k=1	0.4230	0.2015	0.7011	0.2188	0.4924	0.3435	0.3967
top_k=2	0.4100	0.2392	0.5841	0.2316	0.4751	0.3792	0.3865
top_k=3	0.3785	0.2442	0.4785	0.2415	0.4131	0.3314	0.3479
top_k=5	0.3133	0.2289	0.3467	0.2388	0.3174	0.2823	0.2879
Qwen3							
top_k=1	0.5323	0.1574	0.7218	0.2810	0.4698	0.3151	0.4129
top_k=2	0.4963	0.1945	0.5950	0.2883	0.4592	0.3365	0.3950
top_k=3	0.4414	0.2057	0.4846	0.2892	0.4091	0.3205	0.3584
top_k=5	0.3546	0.1960	0.3496	0.2686	0.3186	0.2877	0.2959
GritLM							
top_k=1	0.5105	0.2722	0.8312	0.2890	0.5797	0.3650	0.4746
top_k=2	0.4762	0.3100	0.6473	0.3238	0.5351	0.3799	0.4454
top_k=3	0.4394	0.3008	0.5141	0.3258	0.4592	0.3398	0.3965
top_k=5	0.3573	0.2744	0.3616	0.2971	0.3445	0.3055	0.3234
Autoregressive Models							
SPScanner							
top_k=1	0.4551	0.3232	0.7139	0.0656	0.3862	0.3878	0.3886
top_k=2	0.4732	0.3722	0.7210	0.1220	0.4268	0.4386	0.4256
top_k=3	0.4604	0.3712	0.6434	0.1354	0.4237	0.4188	0.4088
top_k=5	0.4060	0.3469	0.4965	0.1582	0.3893	0.3605	0.3596
AttentionRetriever (Ours)							
LLaMA-3.2 3B							
top_k=1	0.5477	**0.5030**	0.8550	0.2679	**0.6065**	0.3840	0.5274
top_k=2	0.5704	0.4563	0.8394	0.3158	0.6010	**0.4843**	0.5445
top_k=3	0.5584	0.4618	0.8339	0.3526	0.5998	0.4738	**0.5467**
top_k=5	0.5133	0.4270	0.8009	**0.3528**	0.5810	0.4576	0.5221
Qwen-2.5 3B							
top_k=1	0.5583	0.4827	**0.8555**	0.2220	0.5448	0.0891	0.4587
top_k=2	**0.5774**	0.4443	0.8454	0.2930	0.5709	0.3349	0.5110
top_k=3	0.5663	0.4551	0.8422	0.3106	0.5655	0.3215	0.5102
top_k=5	0.5155	0.4165	0.8093	0.3125	0.5468	0.3416	0.4904

Table 4: Comparison of proposed method and baselines on single-document retrieval datasets with different choices of top_k values, where best and second best results are marked in **bold** and underline, respectively.

cantly less input tokens, and also outperforms SPScanner in RAG setting.

F Ablation Studies

To verify the effectiveness of all components in the proposed retrieval model, we conducted ablation studies to measure the improvement brought by each component. We set up three variants of our system that removed attention scoring, embedding scoring, and entity graph, and evaluated the variants on the single-document retrieval datasets. The results are presented in Figure 7. In general, all components in our proposed pipeline brought substantial improvement, where attention scoring shows the most outstanding influence. This result is reasonable because attention scoring helps to

Average Length	RepLiQA 970.50	ConditionalQA 1298.13	NaturalQuestions 2438.86	Qasper 3442.26	QASA 4665.09	LongBench-v2-Retrieval 106025.49
BM25	0.0034	0.0049	0.0047	0.0054	0.0067	0.1005
DPR	0.0585	0.1391	0.0824	0.1192	0.1207	3.4655
ANCE	0.063	0.1412	0.1328	0.1764	0.2595	4.9592
CDE	0.1305	0.3393	0.2442	0.3298	0.4222	10.1549
GTR	0.1466	0.3607	0.2834	0.3913	0.5290	9.6679
GTE-Qwen2	0.4574	0.8310	1.1514	1.4872	2.7483	52.4382
Qwen3	0.5962	1.1245	1.4393	1.8504	3.3642	70.3171
GritLM	0.6338	1.3311	1.8482	2.7332	3.5224	101.7907
SPScanner	0.3790	0.7312	0.8978	1.1500	1.6098	46.4398
AttentionRetriever-LLaMA-3.2 3B	0.9199	1.4303	1.5977	2.1085	2.9402	126.8405
AttentionRetriever-Qwen-2.5 3B	0.9111	1.4051	1.6502	2.4614	2.8444	75.7912

Table 5: Comparison of efficiency of proposed method and baselines on single-document retrieval datasets. The efficiency is measured in average processing time (both indexing and retrieval) in number of seconds for each sample.

	Qasper		MultiFieldQA		NarrativeQA		Average	
	F-1	Avg Token Count	F-1	Avg Token Count	F-1	Avg Token Count	F-1	Avg Token Count
Llama-3.1 8B								
Baseline	0.3145	5026.46	0.5430	6996.87	0.2497	29883.02	0.3691	13968.78
RAG with SPScanner	0.2756	370.25	0.5173	449.22	0.1506	363.54	0.3145	394.34
RAG with AttentionRetriever-Llama	0.2929	392.975	0.5436	416.46	0.1654	322.09	0.3340	377.18
RAG with AttentionRetriever-Qwen	0.2697	382.84	0.5413	432.38	0.1408	350.92	0.3173	388.71
Mistral-7B v0.3								
Baseline	0.2933	5595.38	0.4942	7931.67	0.0724	35277.56	0.2866	16268.20
RAG with SPScanner	0.2485	375.04	0.4402	470.87	0.0997	359.56	0.2628	401.82
RAG with AttentionRetriever-Llama	0.2732	400.39	0.4854	431.09	0.1321	316.245	0.2969	382.57
RAG with AttentionRetriever-Qwen	0.2596	388.83	0.4904	451.57	0.1073	347.56	0.2858	395.98
Qwen-2.5 7B								
Baseline	0.3677	5108.42	0.5085	7205.38	0.1422	29920.98	0.3395	14078.26
RAG with SPScanner	0.2953	352.81	0.4766	438.25	0.1525	341.11	0.3081	377.39
RAG with AttentionRetriever-Llama	0.3220	376.13	0.5062	400.66	0.1746	299.74	0.3343	358.84
RAG with AttentionRetriever-Qwen	0.2927	365.56	0.5274	416.63	0.1532	328.52	0.3244	370.23
GPT-5 mini								
Baseline	0.3142	4974.21	0.4455	6899.97	0.2894	29520.40	0.3497	13798.19
RAG with SPScanner	0.2833	333.16	0.4460	409.96	0.2062	325.47	0.3118	356.20
RAG with AttentionRetriever-Llama	0.2980	355.96	0.4855	377.87	0.2393	284.10	0.3409	339.31
RAG with AttentionRetriever-Qwen	0.2971	346.15	0.4947	393.35	0.2076	312.40	0.3331	350.63

Table 6: Comparison of proposed method and baselines on question answering tasks, where *Baseline* represents using the full context to answer the question, while *RAG* only uses the text chunks retrieved by the specified method to answer the question.

Average Length	QASA 4665.09	Qasper 3442.26	RepLiQA 970.50	ConditonalQA 1298.13	NaturalQuestions 2548.97	LongBench-v2-Retrieval 106025.49	Average
AttentionRetriever	0.5584	0.4618	**0.8339**	**0.3526**	**0.5998**	**0.4738**	**0.5467**
Attention only for sentence scoring	0.5344	**0.4864**	0.8296	0.3372	0.5912	0.3910	0.5283
Embedding only for sentence scoring	0.4753	0.3164	0.7417	0.2598	0.4683	0.4242	0.4476
Removing entity graph	**0.5566**	0.4468	0.7869	0.3371	0.5791	0.4443	0.5251

Table 7: Results of the ablation studies. The best performance for each dataset is marked with **bold**. The ablated settings generally have inferior performance compared to AttentionRetriever, demonstrating the effectiveness of the proposed approach.

model contextual and causal dependencies, two major challenges in long document retrieval.

G Prompts

We used the following prompt for all of our experiments:

> You are an AI assistant. User will you give you a task. Your goal is to complete the task as faithfully as you can.
> Article: {text}
> You are given an article and a question. Answer the question as concisely as you can, using a single phrase or sentence if possible.
> Question: {question}
> Answer:

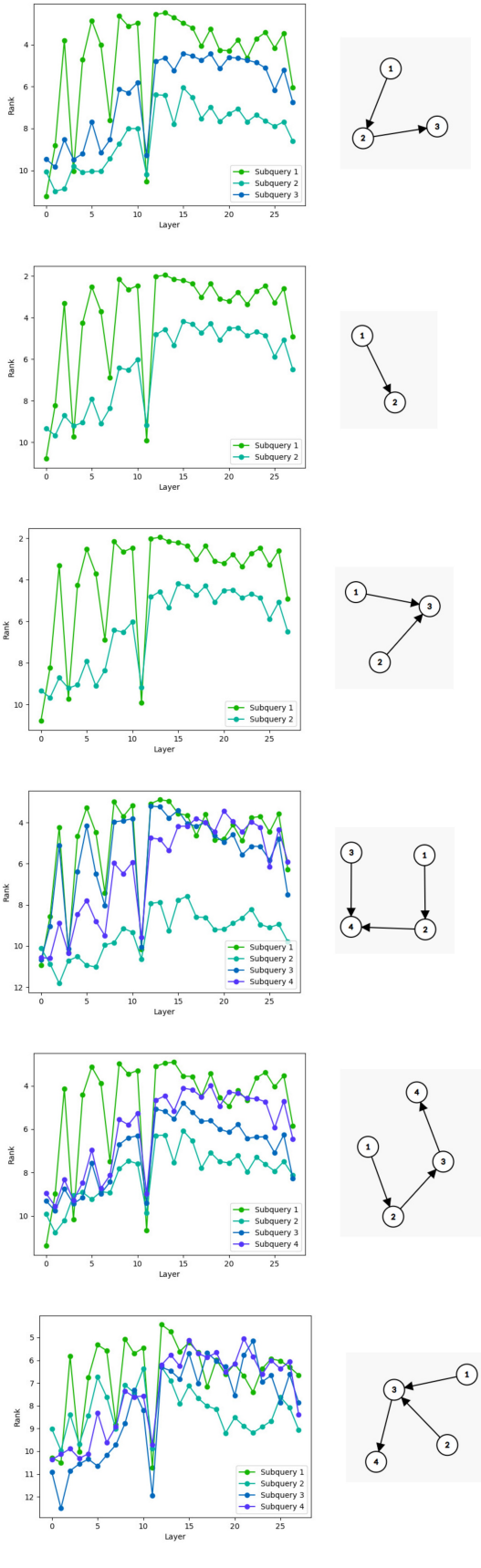


Figure 10: Detailed average ranks for different types of queries.

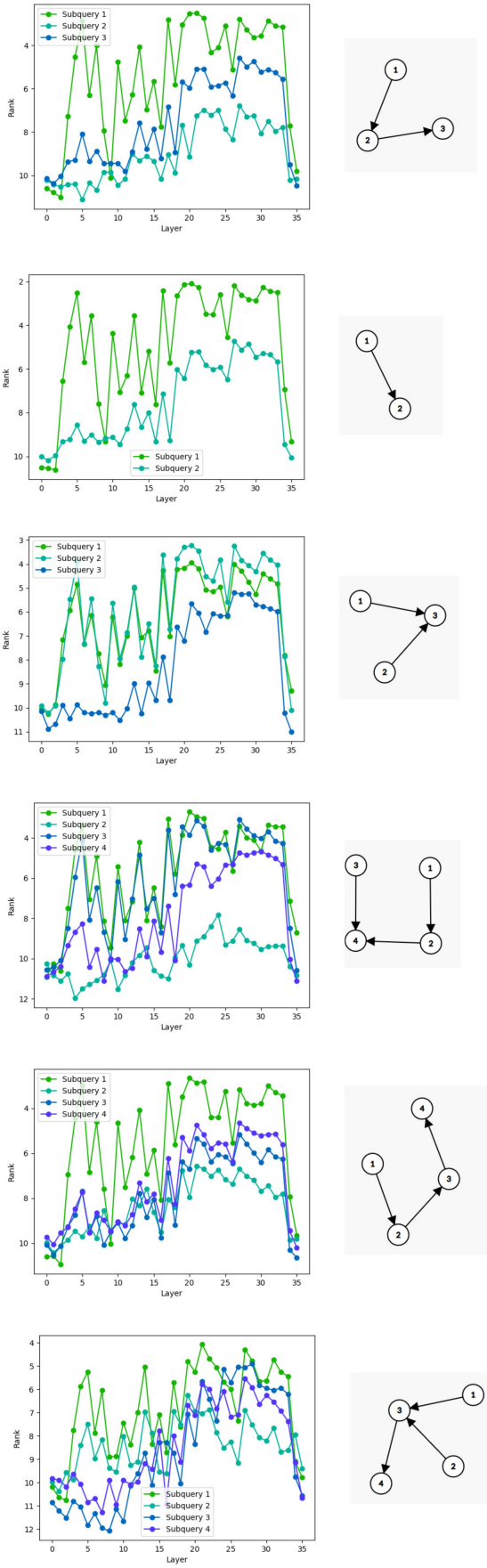


Figure 11: Detailed average ranks for different types of queries.

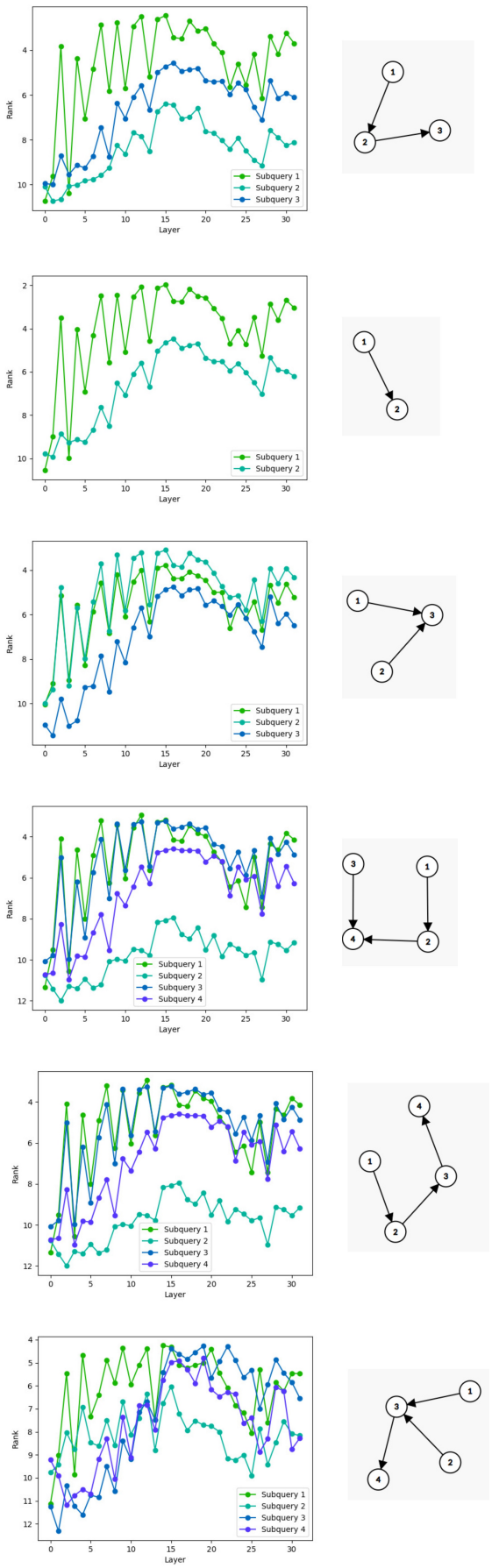


Figure 12: Detailed average ranks for different types of queries.