QRMeM: Unleash the Length Limitation through Question then Reflection Memory Mechanism

Bo Wang[1,3] , Heyan Huang[1,3], Yixin Cao[2], Jiahao Ying[4], Wei Tang[5], Chong Feng[1,3]
[1]Beijing Institute of Technology, China
[2]Fudan University, China
[3]Southeast Academy of Information Technology, Beijing Institute of Technology, China
[4]Singapore Management University, Singapore
[5] University of Science and Technology of China, China
bwang@bit.edu.cn

Abstract

While large language models (LLMs) have made notable advancements in natural language processing, they continue to struggle with processing extensive text. Memory mechanism offers a flexible solution for managing long contexts, utilizing techniques such as compression, summarization, and structuring to facilitate nuanced and efficient handling of large volumes of text. However, existing techniques face challenges with static knowledge integration, leading to insufficient adaptation to task-specific needs and missing multi-segmentation relationships, which hinders the dynamic reorganization and logical combination of relevant segments during the response process. To address these issues, we introduce a novel strategy, Question then Reflection Memory Mechanism (QRMEM), incorporating a dual-structured memory pool. This pool synergizes static textual content with structured graph guidance, fostering a reflective trial-and-error approach for navigating and identifying relevant segments. Our evaluation across multiple-choice questions (MCQ) and multi-document question answering (Multi-doc QA) benchmarks showcases QRMEM's enhanced performance compared to existing approaches. Our code is available at https://github.com/wang202111/qrmem.

1 Introduction

Large language models (LLMs) have achieved significant advancements in natural language processing. Nevertheless, they encounter difficulties handling extensive information, underscoring the necessity for efficient long-context processing. Expanding the context window size is a crucial way to handle the problem, including positional embedding and key-value cache (Beltagy et al., 2020; Zaheer et al., 2020; Chen et al., 2023b). However, the inclusion of more information inevitably results in the accumulation of irrelevant document content.
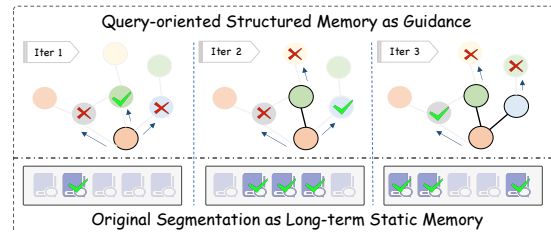

Figure 1: Question then Reflection Memory mechanism is divided into two parts. The upper part, Structured Memory, is a graph constructed from entities and relations extracted from the text. The lower part, Original Segmentation, is the original segmentation. These two parts are linked by the co-occurrence relationships of entities appearing in the segments. QRMEM starts from the core entity of the question and expands on the Structured Memory to find the most relevant next entity node. It uses this to expand and revise the segment sets until the corresponding segment set can answer the question.

Retrieval-based methods aim to filter relevant parts from the original long documents (Zhang et al., 2023; Yu et al., 2023c). They achieve this through various retrieval operations, such as dense, sparse, and reranking, between the text and the query. Another line of approaches, known as memory mechanism methods (Liang et al., 2023), takes the source as a memory pool. These methods offer a solution for managing long contexts and provide an opportunity to handle extraneous documents effectively. Existing memory mechanisms employ several strategies for indexing extensive information. The strategies include storing most recentest acquired information (Liang et al., 2023; Ebbinghaus, 2013), selecting contents based on their relevance, importance, and topics (Zhong et al., 2024; Park et al., 2023), and handling long contexts through techniques such as compression (Pan et al., 2024), summarization (Chen et al., 2023a), and structuring (Hu et al., 2023a).

Despite the existing strategies, several critical issues still persist due to limitations in the accuracy of retrieval modules or other memory construct

tools. First, the memory mechanism is built primarily on human prior knowledge or predefined structure, which may not always align perfectly with the nuanced needs of real-world applications. This results in the gap between the practical scenario queries and the predefined indexing that occurs in the memory mechanism, making it challenging to find relevant information. Additionally, tools or LLMs could be employed to manage extensive information in the memory through processes such as compression, structuring, or other methods, which can lead to error propagation or information loss. Consequently, this can result in biased answers based on memory. Moreover, pinpointing the exact relevant information within these long contexts is inherently challenging. This difficulty is exacerbated by the needs for multi-hop reasoning, long-distance dependencies, and distinguishing between similar segments, which requires tracing information across multiple steps or layers of data, making it difficult to directly locate the segments needed for answering questions.

Socratic inquiry involves asking questions to stimulate critical thinking and illuminate ideas. Inspired by this, we propose a Question then Reflection based Memory Mechanism, which guides thinking through questioning and directs the next steps through reflection on the current state, shown in Figure 1. This mechanism enables the utilization of structured knowledge as guidance to mine the supporting segment. To mitigate the gap between practical scenario queries and predefined indexing, the query-oriented structured memory is constructed by dynamically aligning with the query's needs. The structured memory is formed as a graph with entities and open relations as nodes and edges, which could be a guide to mine the supporting segment. To address information bias during memory construction, we additionally include the original segments as a long-term static memory. These segments are associated with the entity to leverage structured memory for further utilization. Thus, only the filtered segments are used to ensure information fidelity when answering the final question. As for addressing the challenge of pinpointing dispersed relevant segments, we propose three strategies, ENTITY TRIAL achieves comparable results by navigating through only the information of entities and segments in the memory. GRAPH EXPANSION SEARCH uses the relationships between entity nodes as guidance, thereby alleviating complex issues during the retrieval process. QRMEM

reflects on error information from past experiences during multiple iterations, which makes the subsequent steps more effective in finding the relevant segmentation. To summarize, our main contributions are as follows:

- A query-oriented method is proposed to bridge the gap between predefined memory and queries while a dual-structure memory pool is built to mitigate the influence of reorganizing memory.
- Three distinct methodologies are explored utilizing the memory pool, including QRMEM, ENTITY TRIAL, and GRAPH EXPANSION SEARCH. These methodologies adopt different strategies in the interaction process with memory.
- Experimental results on multiple-choice questions (MCQ) and multi-document question answering (Multi-doc QA) show that our model achieves superior results with the various methods, validating the effectiveness of our approach.

2 Methodology

The components of QRMEM are divided into Dual-structure Memory Construction and Reflection-based Relevant Segments Navigation.

2.1 Dual-structure Memory Construction

The approach involves partitioning the memory component into two distinct sections, structured and long-term static memory. The construction is depicted in Figure 2. The structured aspects of the memory resemble human memory anchors, assisting in finding more relevant chunks related to the query as one's understanding deepens. The Structured Memory serves only as guidance to avoid error accumulation during the graph construction process, while the Original Segmentation is used to answer questions, further preventing information loss or misinterpretation.

Query-oriented Graph Initialization: The document D is divided into K segmentation $\{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_K\}$, which is used as the static memory to keep the original information. And the structured memory is constructed by the graph $G_k = \{E_k, R_k\}$, where $\mathbf{E}_k$ is the set of nodes and $\mathbf{R}_k$ is the set of edges. The entities in the original document, which serve as nodes in the graph, are crucial for identifying the relevant segmentation. We first prompt LLM to extract the key entities according to question Q, then combine with the schema-based tool[1] to get the topic entities $\mathbf{E}_k = \{e'_{k,1}, e'_{k,2}, ..., e'_{k,N_k}\}$ in $\mathbf{S}_k$, where N_k is the number of entities in $\mathbf{S}_k$.

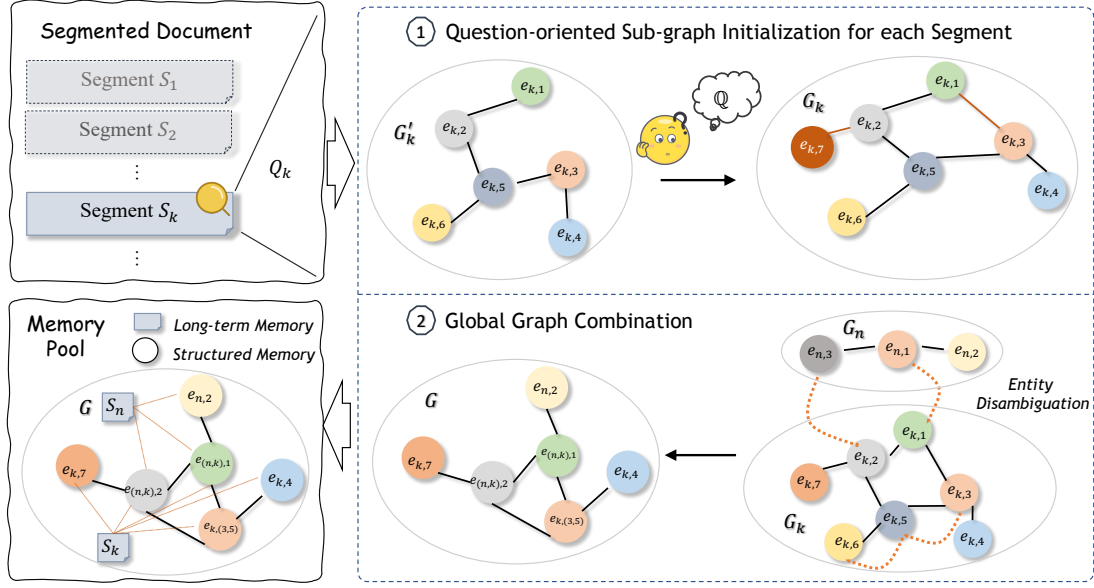[1] https://stanfordnlp.github.io/CoreNLP

Figure 2: Dual-structure Memory Construction pipeline in QRMEM, where e is the entity inside each graph. ① For each segment, the sub-graph G_k is initialized according to the question Q to be answered, then updated by the further generated questions $\mathbb{Q}$ to the current graph. ② Sub-graphs are combined into global graph G through entity disambiguation and relation fusion. All the segmentation $\mathbf{S}_k$ is linked with the entity that appeared to construct the dual-structured memory pool.

The requirement for a fixed schema in Relation Extraction (RE) significantly constrains domain specificity, making it challenging to adapt to complex scenarios. Consequently, we adopt an approach akin to Open Information Extraction (Jia et al., 2022) for relation extraction. Instead of relying on segments from the original document, descriptive relation representations $\{r_{k,(i,j)} \in \mathbf{R}_k\}$ between $e_{k,i}$ and $e_{k,j}$ are generated through LLM. Thus, the edges in the graph G_k are obtained.

Doubt serves as a fundamental catalyst for the acquisition of knowledge. We advance the methodology of knowledge graph updates through a mechanism predicated on question generation. This mechanism is formalized as $\mathbb{Q}_k = \text{LLM}(T, \mathbf{S}_k, \mathbf{E}_k, \mathbf{R}_k)$, where $\mathbb{Q}_k$ denotes the generated question leveraging a Large Language Model (LLM) and the T represents the summary of the entire document. To ensure the diversity of the generated questions, a novel question is incorporated into the question pool only if its ROUGE-L similarity score with any existing question is less than 0.6.[2] Entities and relations are further supplemented through similar techniques based on the questions $\mathbb{Q}_k$ during the initial stage.

Global Graph Combination: In the process of integrating sub-graphs $\{G_1, G_2, ..., G_K\}$ into a comprehensive global graph G, we further apply the entity disambiguation and relation merging. After entity disambiguation, entities associated with distinct segments are further elaborated into triples of the form $e_{k,j} = \{e'_{k,j}, \mathbb{M}_{k,j}, \mathbb{N}_{k,j}\}$, thereby facilitating the update of paragraph indices. Here, $\mathbb{M}_{k,j}$ and $\mathbb{N}$ represent the sets of entity mentions and segment indices, respectively. After merging entity pairs $(e_{n,i}, e_{k,i'})$ and $(e_{n,j}, e_{k,j'})$ into consolidated entities $e_{(n,k),i}$ and $e_{(n,k),j}$, Following the initial setup phase, we generate questions for merging relationships Q_s according to the entities and their original segments, which could benefit from rethinking the relation between entity pairs. Consequently, relations $r_{n,(i,j)}$ and $r_{k,(i',j')}$ are amalgamated into a unified relation $r_{(k,n),(i,j)} = \text{LLM}(r_{n,(i,j)}, \mathbf{S}_n, r_{k,(i',j')}, \mathbf{S}_k, Q_s, Q, T)$. Static segments will be associated with the entity nodes, thereby organizing the connection between the structured and long-term parts in the memory pool, as shown in Figure 2.

2.2 Reflection-based Segments Navigation

To accurately determine the segments relevant to question Q, three segment-searching algorithms are proposed to explore the impact of different focuses. The approaches begins by extracting the initial set of entities $\mathbb{E}_s$ and the associated segments $\mathbb{R}_s$ from the query. Subsequently, segments corresponding

---

[2]Excessive number of questions could introduce further noise, we empirically selected 0.6 as threshold.
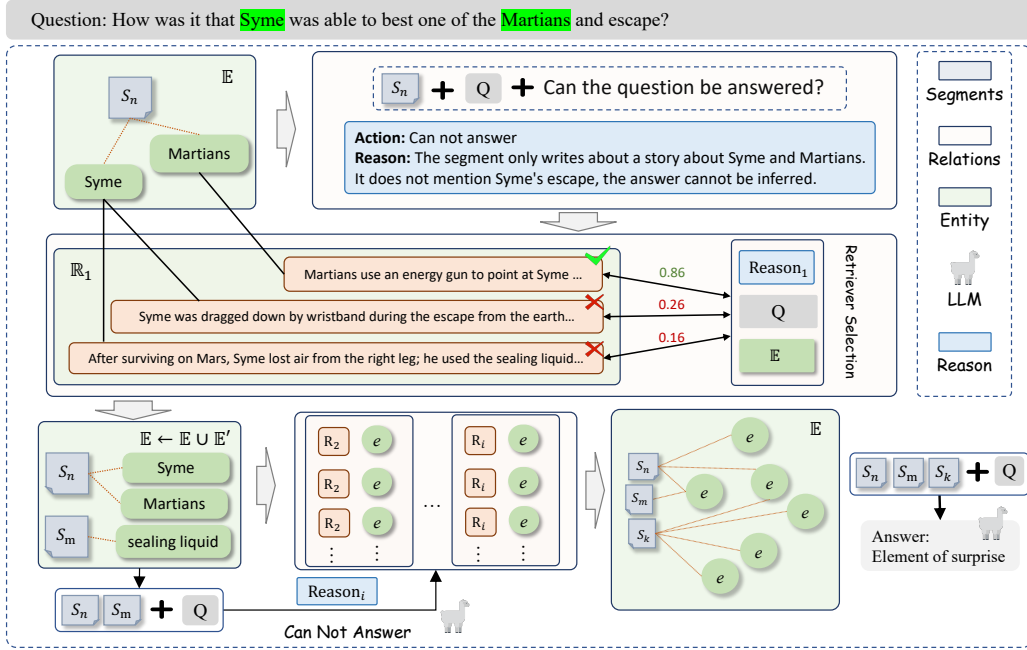
Figure 3: Strategies for KNOWLEDGE-DIRECTED NAVIGATION in QRMEM. Navigation starts with a subgraph containing only the target entity from the question and its linked segments. The subgraph is then expanded based on the relevance of relationship edges connected to the current entity. The process will continue until the linked segments are sufficient to answer the question.

to the initial entity set $\mathbb{E}_s$ are extracted and designed as important segments $\mathbb{S}_{\text{imp}}$.

ENTITY TRIAL. To enable multiple explorations in complex scenarios and allow the model to find more relevant segments based on previous iterations, we employ a straightforward ENTITY TRIAL approach. This method iteratively updates the entity set without the guidance of relations to enhance the relevance of text segments, using a bipartite graph structure composed solely of entities and segment nodes in the memory pool. After obtaining the initial answer from the model, we utilize its reasoning capability to verify whether the current entity set $\mathbb{E}$ can sufficiently answer the question. If the model cannot answer the question, it iterates by updating $\mathbb{E} \leftarrow LLM(\mathbb{E}, R, \mathbb{S}_{\text{imp}}, Q)$. This process continues until the model can answer the question or reaches the current window length limit.

GRAPH EXPANSION SEARCH. As the number of nodes increases, the model's ability to filter nodes and the window length both fall short of meeting the need of ENTITY TRIAL. GRAPH EXPANSION SEARCH further leverages the structured part of the memory pool, G, to enhance exploration from segments. $\mathbb{E}_s$ is first expanded by incorporating adjacent nodes within the graph. The expansion is guided by the computation of similarity scores between the given query Q and the $\mathbb{R}_s$, retaining nodes $\mathbb{E}'$ with high similarity. The entity set $\mathbb{E}$ is

updated through $\mathbb{E} \leftarrow \mathbb{E}' \cup \mathbb{E}$ following each expansion iteration. The expansion process continues until either a predefined threshold of expansion iterations is reached or no new edges with sufficiently high similarity scores are found. The resultant expanded set of entities $\mathbb{E}$ is then to generate the elaborated query through $\mathbb{Q} = LLM(\mathbb{E}, \mathbb{R}, Q)$. $\mathbb{Q}$ is utilized to retrieve relevant segments, which are constructed into c to subsequently address Q. c is the concatenation of the original question and the generated question used to retrieve all relations.

QRMEM. To trace the experience across multiple iterations, QRMEM utilizes a reflection-based navigation strategy that dynamically traverses the graph G, which is shown in Figure 3. Initially, QRMEM sets up an additional segmentation set, $\mathbb{S}_{\text{add}}$, as $\emptyset$ for further navigation. For each iteration, the real relevant segments $\mathbb{S}_{\text{mix}}$ is formed by combination of $\mathbb{S}_{\text{imp}}$ and $\mathbb{S}_{\text{add}}$. Then the LLM is employed to evaluate whether $\mathbb{S}_{\text{mix}}$ can answer the question Q by the feedback of the question Q and the reason R for the failure is identified if not. While the $\mathbb{S}_{\text{mix}}$ is insufficient for the current query, the first-order adjacent nodes $\mathbb{E}_{\text{adj}}$ of the current entities and all connecting edges $\mathbb{R}$ are collected. Drawing upon the lessons learned from the previous iteration, the query, reason, and current entities $[Q; R; \mathbb{E}_s]$ are used to select the next expanded node. Contriever(Izacard et al., 2022) is

used for calculating the similarity score between the $[Q; R; \mathbb{E}_s]$ and each edge in $\mathbb{R}$. After the most relevant adjust node e is selected and merged with the new entity e into the entity set $\mathbb{E}$, the segments corresponding to the new entity e are collected. All segments not present in $\mathbb{S}_{mix}$ are added to $\mathbb{S}_{add}$ with the similarity score. Allowing for thorough updates to $\mathbb{S}_{add}$, a crucial step is to ensure that the context window remains sufficient after updates. $\mathbb{S}_{add}$ is processed and filtered according to the similarity score while adhering to the window constraints. This iterative segment combination, window checking, question evaluation, and sub-graph expansion process continues until the question Q is satisfactorily answered. The detail of the algorithm is shown in Algorithm 1.

3 Experiment

3.1 Baselines

To evaluate the effectiveness of the proposed method, we compare the QRMEM with several methods. **Retrieval Methods**. We use dense retrievers Contriever (Izacard et al., 2022) and BM25 for all datasets. **Vanilla**, we use the model with 4096 context windows, including LLaMa-70BChat (Touvron et al., 2023) and GPT-3.5-turbo. **Summary**, we summarize the long context to fit the context window size. **DECOM**(Fu et al., 2021) simplifies complex questions into sub-questions, answers them sequentially. **LLM-Lingua** (Pan et al., 2024), compresses prompts to speed up inference and enhance response efficiency in large language models. **Long context model**, including Vicuna-7b-32k, Vicuna-13b-16k, Longchat-7b-32k, Longchat-13b-16k,Llama-70b-32k. **MEMWALKER** (Chen et al., 2023a), which utilize a tree of summary nodes.

3.2 Experiment Setting

The official LLaMa-70B-Chat and GPT-3.5-turbo are used for the majority of our experiments. The two models both have a maximum 4,096 token context length. We utilize the model in a zero-shot prompting fashion without further fine-tuning or using few-shot examples for our tasks. For both memory tree construction and generating action and reasoning for navigation, we use top-p sampling. The temperature is initially set to 0, but if the output cannot be parsed, we increase the temperature to 0.7 and retry up to four times. The segment size is set to 600 for all datasets.

3.3 Dataset & Metrics

The evaluation datasets predominantly comprise sections on multiple-choice questions (MCQ) and multi-hop question-answering (Multi-Doc QA). **QuALITY** (Pang et al., 2022) is employed for the MCQ component, which comprises extended narratives derived from Project Gutenberg, coupled with questions annotated by expert human annotators. For our experimental analysis, we selected a subset consisting of 222 examples, and accuracy is used for evaluation. To evaluate the capability of short-context models in comprehending extended texts, we opted against using the original multi-hop QA dataset. Instead, we employed **Multi-Doc QA** derived from LongBench (Bai et al., 2023), with an average length exceeding 8,000 tokens, ensuring suitability for evaluating the understanding of extensive textual information. The datasets incorporate the Wikipedia passages encompassed version of HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020) and MuSiQue (Trivedi et al., 2022). Consistent with previous works (Yang et al., 2018; Bai et al., 2023), exact match (EM) and F1 is selected for evaluation.

3.4 Main Result

We evaluate the proposed model QRMEM under the long context question-answer setting. Table 1 shows the comparison results of QRMEM with different navigation methods on QuALITY and three Multi-doc QA datasets against all baseline methods. ENTITY TRIAL is only tested on QuALITY due to the excessive number of entities present on Multi-Doc QA. We can see that:

1) Our study presents significant advancements over retrieval methods on the QuALITY dataset, on both the LLaMa-70B-Chat and GPT-3.5-turbo models. Detailed results related to retrieval methods are referred to in Appendix B. Notably, our approach, when compared to models utilizing longer context windows, outperforms other models designed for long-text comprehension, including GPT-3.5-16k. This highlights the effectiveness of our method in selecting relevant segments, promoting a comprehensive understanding of complex documents.

2) Furthermore, our methods demonstrated exceptional performance on Multi-Doc QA datasets. Specifically, on the 2WikiMultihopQA and MuSiQue datasets, our approach achieved remarkable results and yields comparable results to existing retrieval models on the HotpotQA dataset.

Method	QuALITY	HotpotQA		2WikiMultihopQA		MuSiQue	
	Acc	EM	F1	EM	F1	EM	F1
Long Context Model							
Vicuna-7b-16k	0.4175	0.2750	0.2316	0.2750	0.1779	0.0700	0.0928
Vicuna-13b-16k	0.5191	0.2800	0.3074	0.3400	0.2674	0.1150	0.1315
Longchat-7b-32k	0.2938	0.3750	0.2932	0.3750	0.2012	0.1400	0.1256
Longchat-13b-16k	0.3250	0.3000	0.2465	0.2850	0.1945	0.1100	0.1335
Llama-2-70b-32k	<u>0.5765</u>	**0.4000**	<u>0.3372</u>	**0.4200**	<u>0.3725</u>	**0.1750**	<u>0.1605</u>
GPT-3.5-16k	**0.7162**	<u>0.3900</u>	**0.4415**	<u>0.4050</u>	**0.4278**	<u>0.1550</u>	**0.1888**
LLaMa-70B-Chat							
BM25 (top 3)	0.6055	**0.4800**	**0.4418**	0.4000	0.3285	0.1750	0.1697
Contriever (top 3)	0.6098	0.3900	0.3552	0.4500	**0.3632**	0.1300	0.1640
Vanilla (keep left)	0.6131	0.3750	0.2966	0.4050	0.2393	0.1000	0.0898
Vanilla (keep right)	0.6251	0.2950	0.2655	0.3400	0.2185	0.1200	0.1202
MEMWALKER†	0.6302	0.4350	0.2046	0.3850	<u>0.3462</u>	0.1600	0.1628
LLMLINGUA	0.4636	0.2750	0.2662	0.2750	0.2439	0.0600	0.0334
DECOM	0.4515	0.4250	0.3938	0.3350	0.3136	0.1300	0.1277
QRMEM (TE)	0.6351	-	-	-	-	-	-
QRMEM (GES)	<u>0.6441</u>	0.4550	0.4032	<u>0.4650</u>	0.3044	<u>0.1800</u>	<u>0.1933</u>
QRMEM (KN)	**0.6531**	<u>0.4650</u>	<u>0.4211</u>	**0.4700**	0.3314	**0.2000**	**0.1996**
GPT-3.5-turbo							
BM25 (top 3)	0.6359	0.3850	0.4511	0.3200	0.3471	0.0850	0.1476
Contriever (top 3)	0.6275	0.3650	0.3950	0.4100	<u>0.4523</u>	0.1350	<u>0.1948</u>
Vanilla (keep left)	0.6448	0.3650	0.4229	0.3350	0.3615	0.0650	0.1020
Vanilla (keep right)	0.6687	0.2550	0.3098	0.2550	0.2691	0.0750	0.1213
MEMWALKER†	0.6622	0.3350	0.4113	0.4050	0.4432	0.0950	0.1323
LLMLINGUA	0.5360	0.3200	0.3393	0.2600	0.2817	0.1000	0.0811
DECOM	0.5495	0.3850	0.3933	0.3750	0.3939	0.1000	0.1086
QRMEM (TE)	<u>0.6936</u>	-	-	-	-	-	-
QRMEM (GES)	0.6801	<u>0.4150</u>	**0.4731**	<u>0.4250</u>	0.4432	<u>0.1850</u>	0.1931
QRMEM(KN)	**0.7207**	**0.4200**	<u>0.4632</u>	**0.4250**	**0.4698**	**0.1900**	**0.2094**

Table 1: Mian results on QuALITY and Multi-doc QA datasets, †denotes that the results are reproduced based on the original paper. TE denotes the ENTITY TRIAL, GES denotes the GRAPH EXPANSION SEARCH, and KN denotes the QRMEM. We bold the highest scores and underline the second-highest scores for each block.

Even though BM25 obtained a satisfactory result on HotpotQA, it still struggles with complex tasks like QuALITY and MuSiQue. Across all three datasets, our method outperformed long-document comprehension models. This indicates our approach's adaptability to multi-hop tasks that require the understanding of entity relationships, capturing these relationships more effectively than long-document models. The lesser improvement on fewer inference steps HotpotQA questions and more significant advancements on the complex MuSiQue and QuALITY tasks illustrate our method's effectiveness in organizing original paragraphs for complex tasks. GPT-3.5-16k tends to focus more on the overall context, while GPT-3.5-turbo pays closer attention to details. This is also why GPT-3.5-16k shows better performance on QuALITY datasets.

3) Three strategies proposed in our study outperformed retrieval-enhanced and direct truncation strategies. ENTITY TRIAL's performance improvement demonstrates the effectiveness of iterative exploration. By using entities as anchors and leveraging the reflective iterative exploration of segments, it shows the impact of reflective iteration in acquiring relevant segments. GRAPH EXPANSION SEARCH, although lacking an iterative navigation process, still utilizes structured memory to guide the LLM. This indicates that our query-based structured memory provides better guidance and relevance compared to predefined indexing. QRMEM achieves the best performance. It combines the strengths of both ENTITY TRIAL and GRAPH EXPANSION SEARCH, using the graph to guide the model and leveraging past errors to inform future steps. This approach allows the model to utilize specific structured knowledge while retaining the ability to iteratively combine different segments.

3.5 Ablation Study

In this section, we study the effectiveness of each proposed module by removing them from QRMEM on QuALITY Datasets. Results in Table 2 show that removing any component decreases performance, which observe their individual contri-

Method	Overall	Easy	Difficult
QRMEM	0.7207	0.7000	0.7450
-w/o GRAPH UPDATE	0.6846	0.7166	0.6470
-w/o OPEN ENTITY	0.6621	0.7083	0.6078
-w/o REFLECTION	0.6801	0.6833	0.6764
-w/o NAVIGATION	0.6351	0.6750	0.5882

Table 2: Ablation study on Quality dataset through GPT-3.5-turbo. Quality is annotated with difficult levels, and we report the results on different levels.

butions to performance. GRAPH UPDATE denotes the additional question generation and graph update during initialization. OPEN ENTITY refers to the entity recognition method through LLM. When these two components are removed, we observed improved performance on the easy level but a significant decline in performance on the difficult level. This phenomenon indicates that query-oriented structured memory is crucial for addressing complex problems. REFLECTION involves adding reasoning as a reflection when calculating similarity with relations. Without leveraging past experiences and only relying on the query to traverse the graph, there is a significant loss in performance. This demonstrates that our reflection mechanism provides more precise guidance. NAVIGATION refers to the navigation stage. Without NAVIGATION, we simply use the start entity to locate the segments and then answer the query. This approach loses the iterative navigation process, resulting in an overall decline in performance. Our technique of reorganizing paragraphs helps to mitigate this issue and compensates for this deficiency.
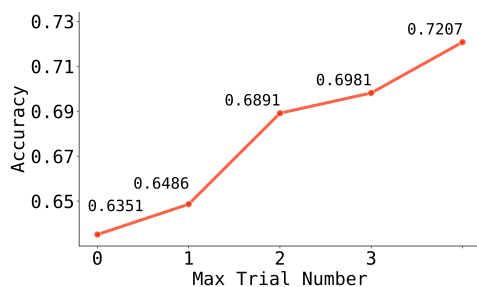
3.6 Performance over Different Max Trials

Figure 4: Accuracy on QuALITY datasets with different max trials number.

In this study, we introduce a constraint on the number of updates allowed for entity set $\mathbb{E}$ during the search process, which is shown as Figure 4. Once the maximum number of updates is reached, the search is terminated, then the answer is gener-

ated based on the current state of $\mathbb{E}$. Our results indicate a significant enhancement in performance transitioning from the first to the second update, underscoring the effectiveness of the navigation strategy. Upon undergoing a series of iterative trial and error processes, it has been observed that the performance of the model reaches a plateau, indicating a deceleration in the rate of improvement. The model could still gain profit from the navigation. Experimental observations indicate that trail number could be a trade-off between efficiency and effectiveness.

3.7 The impact of irrelevant information

Datasets	Head		Middle		Tail	
	EM	F1	EM	F1	EM	F1
Vicuna-7b-16k						
Hotpot	0.3150	0.2283	0.2650	0.1889	0.3250	0.2257
2Wiki	0.3850	0.2387	0.2700	0.1749	0.3150	0.2043
Musique	0.0750	0.0896	0.0500	0.0733	0.1950	0.2060
Vicuna-13b-16k						
Hotpot	0.3750	0.3444	0.3050	0.2643	0.3650	0.3282
2Wiki	0.3400	0.2584	0.3550	0.2864	0.3900	0.3052
Musique	0.0900	0.0783	0.0700	0.0680	0.1000	0.0725

Table 3: Result on Multi-Doc QA datasets, the supporting segments are positioned at the beginning, middle, and tail of the document.

To further investigate the influence of irrelevant documents, we conducted additional tests on cross-document multi-hop questions. In this section, we retained the original wiki support text from the Long Bench dataset without chunking the long text, enabling us to more clearly identify the impact of irrelevant documents. Following (Liu et al., 2023), we investigated the impact of varying the placement of supporting segments within a document, and the result is shown in Table 3. Specifically, we positioned the corresponding support segments at the beginning, middle, and tail of the overall document. Our findings are consistent with those reported by Liu et al. (2023), indicating that the placement of information can significantly affect the model's performance on the Multi-Doc QA task. Therefore, in our subsequent experiments, we placed the multi-hop support text at both the beginning and end of the document, ensuring that the supporting segments remain in these positions to avoid any positional impact.

Irrelevant documents utilized to increase the distance are retained, represented as $[S_1; S_{non}; S_2]$, where S_1 and S_2 are the supporting segments, and S_{non} represent sequences of non-relevant segments. The outcomes are depicted in Figure 5. The results
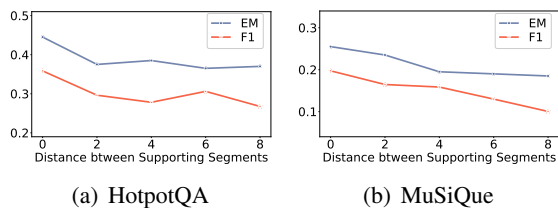
(a) HotpotQA   (b) MuSiQue

Figure 5: The performance over different distances between the supporting segments.

indicate that although the model exhibits heightened attention to the positions at the beginning and end of the text, its performance is still impacted by the distance between documents. Effectiveness consistently declines as distance increases on both the HotpotQA and MuSiQue datasets.

In summary, tasks of increasing complexity necessitate the aggregation of relevant documents, where the effects can even surpass the influence of the locations of the supporting documents. Long context understanding requires high capability in capturing long-distance dependencies that demand complex reasoning. The case study in Appendix C illustrates our ability to find relevant documents in complex situations. QRMEM constructs structured knowledge for each segment, mitigating the impact of long-distance dependencies. By utilizing structured knowledge to reorganize the segments, our approach facilitates the close integration of text with dependency relationships, which leads to superior effectiveness on Multi-Doc tasks compared to the long context windows model.

4 Related Work

4.1 Memory Mechanism

Except for the LLMs with extended context window size (Press et al., 2022; Guo et al., 2022; Ainslie et al., 2023) and vanilla retrieve augment generation methods (Shi et al., 2023; Zhang et al., 2023; Yu et al., 2023b; Chevalier et al., 2023), numerous attempts have been made to enhance the memory capabilities of neural models(Chen et al., 2023a). Memory mechanisms could be predefined representations, including raw natural languages (Zhong et al., 2024; Hu et al., 2023a), structured information (Chen et al., 2023a) such as tuples, databases, etc., and paramterized representations (Yu et al., 2023a; Tang et al., 2024). To leverage the constructed memory, the method stores all the information of the agent-environment interaction history based on long-context strategies, maintains the most recently acquired memories (Liang et al.,

2023; Packer et al., 2023; Ebbinghaus, 2013), and selects memory contents based on their relevance, importance, and topics (Zhong et al., 2024; Park et al., 2023). However, these memory-based algorithms do not allow the preparation of specific memory structures for different queries, and further memory integration is also ignored. We use the Knowledge-directed Navigation method to reorganize the context adjustably.

4.2 KG augmented Language Modeling

Knowledge Graphs (KGs) offer dynamic, structured knowledge representation, enhancing Large Language Models (LLMs) through explicit structured knowledge, as demonstrated by early studies such as (Zhang et al., 2019; Peters et al., 2019; Yamada et al., 2020; Wang et al., 2021b,a; Hu et al., 2023b). Methods include generating SPARQL query backbones (Li et al., 2023a), sampling entity-related triples (Baek et al., 2023), decomposing (Li et al., 2023b), and employing a retriever-reader-verifier(Wang et al., 2023). Other approaches like (Jiang et al., 2023) enable LLMs to navigate KGs through greedy search, indicating a trend towards interactive KG exploration and reasoning (Sun et al., 2023), graph-driven context retrieval (Banerjee et al., 2024), and the formulation of examinations to assess LLM comprehension (Wan et al., 2024). Previous methods primarily incorporate graphs into the response in various ways or introduce external knowledge graphs. Our approach directly utilizes the original documents to construct graphs and employs the original paragraph information during the answer generation process. This strategy effectively circumvents the propagation of errors that may arise during the extraction process.

5 Conclusion

In this study, we propose the Question-then-Reflection Memory mechanism, which enables the utilization of structured knowledge to guide the mining of supporting segments. QRMEM aligns with the query's needs by using structured memory for guidance, and original segments are included as long-term static memory to avoid information bias. We introduce ENTITY TRIAL, GRAPH EXPANSION SEARCH, and QRMEM as methodologies to navigate through structured memory and find relevant segments. QRMEM reflects on errors from past experiences during multiple iterations, allowing the model to learn from its mistakes

and improve performance. Experimental results on MCQ and Multi-doc QA demonstrate that our model achieves superior results with various methods, validating the effectiveness of our approach.

6 Limitations

In this paper, we propose a Question then Reflection Memory Mechanism method for iteratively select the relevant segments. The limitations of the proposed method are as follows: 1) The computational cost of initializing and updating the knowledge in processing the long context, may restrict its system applicability in scenarios where there is a need to quickly build additional knowledge bases. 2) Given the constraints imposed by the size of the model window, reliance on entity resolution is utilized to bridge distinct paragraphs during the extraction of pertinent knowledge in constructing the structure knowledge. The segmentation of text into paragraphs potentially impacts the efficacy of connections within the graph's structure.

In future work, we plan to mitigate the method's computational cost by developing more efficient pipelines. We aim to limit the updates of the graph to a very small local range to reduce the cost of updating the graph structure or explore the incorporation of external knowledge sources to mitigate this limitation and enhance the model's performance.

7 Ethics Statement

This work was conducted in rigorous compliance with the ACL Ethics Policy. All datasets and large language models (LLMs) used for evaluation are publicly available. Furthermore, our work strives to explore a Question and Reflection Memory Me for long-context understanding. We do not foresee any form of negative ethical impact induced by our work.

8 Acknowledgement

References

Joshua Ainslie, Tao Lei, Michiel de Jong, Santiago Ontañón, Siddhartha Brahma, Yury Zemlyanskiy, David C. Uthus, Mandy Guo, James Lee-Thorp, Yi Tay, Yun-Hsuan Sung, and Sumit Sanghai. 2023. Colt5: Faster long-range transformers with conditional computation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5085–5100. Association for Computational Linguistics.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering.

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. 2023. Longbench: A bilingual, multi-task benchmark for long context understanding.

Somnath Banerjee, Amruit Sahoo, Sayan Layek, Avik Dutta, Rima Hazra, and Animesh Mukherjee. 2024. Context matters: Pushing the boundaries of open-ended answer generation with graph-structured knowledge context.

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

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023a. Walking down the memory maze: Beyond context limit through interactive reading.

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

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3829–3846. Association for Computational Linguistics.

Hermann Ebbinghaus. 2013. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 20(4):155.

Ruiliu Fu, Han Wang, Xuejun Zhang, Jun Zhou, and Yonghong Yan. 2021. Decomposing complex questions makes multi-hop QA easier and more interpretable. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 169–180, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mandy Guo, Joshua Ainslie, David C. Uthus, Santiago Ontañón, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. Longt5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 724–736. Association for Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023a. Chatdb: Augmenting llms with databases as their symbolic memory. *CoRR*, abs/2306.03901.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2023b. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*.

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

Shengbin Jia, E. Shijia, Ling Ding, Xiaojun Chen, and Yang Xiang. 2022. Hybrid neural tagging model for open relation extraction. *Expert Syst. Appl.*, 200:116951.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. 2023a. Few-shot in-context learning on knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6966–6980. Association for Computational Linguistics.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Lidong Bing, Shafiq Joty, and Soujanya Poria. 2023b. Chain of knowledge: A framework for grounding large language models with structured knowledge bases.

Xinnian Liang, Bing Wang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. 2023. Unleashing infinite-length input capacity for large-scale language models with self-controlled memory system. *CoRR*, abs/2304.13343.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *CoRR*, abs/2307.03172.

Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph E. Gonzalez. 2023. Memgpt: Towards llms as operating systems. *CoRR*, abs/2310.08560.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression.

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

Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST 2023, San Francisco, CA, USA, 29 October 2023- 1 November 2023*, pages 2:1–2:22. ACM.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph.

Wei Tang, Yixin Cao, Jiahao Ying, Bo Wang, Yuyue Zhao, Yong Liao, and Pengyuan Zhou. 2024. A + b: A general generator-reader framework for optimizing llms to unleash synergy potential.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Fanqi Wan, Xinting Huang, Leyang Cui, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Mitigating hallucinations of large language models via knowledge consistent alignment.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering.

In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023a. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023b. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023c. Augmentation-adapted retriever improves generalization of language models as generic plug-in. *arXiv preprint arXiv:2305.17331*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Merging generated and retrieved knowledge for open-domain qa. *arXiv preprint arXiv:2310.14393*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19724–19731. AAAI Press.

A Navigation strategy

To better describe our navigation strategy, the detailed algorithm of navigation is shown as Algorithm 1

Algorithm 1 Reflection-Based Navigation Strategy

Require: Query Q, Initial Entity Set $\mathbb{E}_s$, Initial Relation Set $\mathbb{R}_s$, Segment Set $\mathbb{S}_{add} \leftarrow \emptyset$, Context Windows Size L

Ensure: Answer to the question Q

1: $\mathbb{S}_{imp} \leftarrow$ Extract segments corresponding to $\mathbb{E}_s$
2: $\mathbb{S}_{mix} \leftarrow \mathbb{S}_{imp} \cup \mathbb{S}_{add}$
3: **while** not answered(Q) **do**
4:    **if** LLM can answer Q with $\mathbb{S}_{mix}$ **then**
5:       **return** Answer
6:    **else**
7:       $R \leftarrow$ LLM$(\mathbb{S}_{mix}, Q)$
8:       $\mathbb{E}_{adj} \leftarrow$ First-order adjacent nodes of $\mathbb{E}_s$
9:       $\mathbb{R} \leftarrow$ Connecting edges of $\mathbb{E}_{adj}$
10:      $S \leftarrow$ Similarity$([Q; R; \mathbb{E}_s], \mathbb{R})$
11:      $e \leftarrow$ Max$_S(\mathbb{R})$
12:      $\mathbb{E} \leftarrow \mathbb{E} \cup \{e\}$
13:      $\mathbb{S}_{new} \leftarrow$ Extract segments corresponding to e
14:      $\mathbb{S}_{add} \leftarrow \mathbb{S}_{add} \cup (\mathbb{S}_{new} \setminus \mathbb{S}_{mix})$
15:    **end if**
16:    $\mathbb{S}_{mix} \leftarrow \mathbb{S}_{imp} \cup \mathbb{S}_{add}$
17:    **if** LEN$(\mathbb{S}_{mix}) > L$ **then**
18:      $\mathbb{S}_{add} \leftarrow$ Filter $\mathbb{S}_{add}$ according to S
19:      $\mathbb{S}_{mix} \leftarrow \mathbb{S}_{imp} \cup \mathbb{S}_{add}$
20:    **end if**
21: **end while**

B Retrieval Result

We progressively incorporated the content retrieved into the original document in the retrieval experiments. "Max" indicates including retrieved content in the prompt until the length limit is reached. For the multi-doc QA dataset, we reported Exact Match (EM), and for the QuALITY dataset, we reported Accuracy. The results are shown in the table 4. Since performance does not improve with the inclusion of more data, we selected the top 3 with the higher average values for comparison in our paper.

Table 4: Results about the Retrieval Methods

Method	Hotpot	2wiki	Musique	Quality
BM25(top 1)	0.3800	0.2450	0.1150	0.5288
BM25(top 2)	0.4450	0.3600	0.1500	0.5705
BM25(top 3)	0.4800	0.4000	0.1750	0.6055
BM25(Max)	0.4550	0.4050	0.1750	0.6208
Contriever(top 1)	0.3350	0.2800	0.1000	0.5292
Contriever(top 1)	0.3850	0.3950	0.1600	0.5959
Contriever(top 3)	0.3900	0.4600	0.1300	0.6098
Contriever(max)	0.4300	0.4450	0.1550	0.6385
RKDN(ours)	0.4650	0.4700	0.2000	0.6531

C Case Study

As illustrated in Table 5, when crucial entities are absent in the query, we can employ an iterative navigation process to search for and identify the relevant entities gradually. Although irrelevant segments may be included, the noise introduced is less than that of retrieval methods. The first two segments rank lower in retrieval, indicating that for complex reasoning tasks, it is essential to construct memory based on the query and reorganize the text to bridge the gap with the query. This validates the effectiveness of QRMEM.

Segments	Rank	Contain
Valencia CF had a successful season, finishing in the top four of La Liga and thus qualifying for the UEFA Champions League, thanks to the extension of the competition to include more teams from the top leagues. Valencia also won the Copa del Rey, ending a long trophy drought and marking a successful end to Italian coach Claudio Ranieri's first spell at the club.	7	✔
José Daniel Valencia (born 3 October 1955) is an Argentine former professional footballer who played as an attacking midfielder. He is perhaps most famous for having been part of the 1978 World Cup winning squad.Valencia started his club career at Gimnasia y Esgrima de Jujuy but was soon transferred to Talleres de Córdoba, the club at which he would play most of his career. At Talleres, Valencia suffered the disappointment of finishing runner-up in Nacional 1977, finishing third in Metropolitano 1980, and losing the semi-finals on four occasions.	5	✔
Higinio Ortúzar Santamaría (10 January 1915 – 8 November 1982) was a Chilean footballer who made his entire career in Spain. The first Chilean in Spanish football, he made his debut for Erandio Club in 1935, and next he played for Barakaldo CF, Athletic Bilbao, Valencia CF, Real Valladolid, and Real Sociedad. He was loaned to Racing de Santander in 1936 for 4,500 pesetas, but he couldn't play due to the Spanish coup of July.	1	✘
José Raúl Aveiro Lamas (born 18 July 1936) is a Paraguayan former professional footballer who played as a striker. Born in Asunción, Aveiro played for Sportivo Luqueño, Valencia, Valencia Mestalla, Elche, Ontinyent, and Constància. He was also a member of the Paraguay national team between 1957 and 1959.	2	✘
Claudio Javier López (Spanish pronunciation:, born 17 July 1974) is an Argentine former footballer, who played as a forward. Nicknamed Piojo (louse), he is best known for his spells with Valencia in Spain and Lazio in Italy. López also had a notable impact in the Argentina national team, participating in two World Cups.	3	✔
Question: Which retired Argentine footballer who played as a forward was a main player for Valencia CF? **Golden Answer**: Claudio Javier López **Retriever Answer**: Mario Alberto Kempes Chiodi **Compressed Answer**: Mario Alberto Kempes Chiodi QRMEM **Answer**: Claudio Javier López		

Table 5: Case studies of solving Mulri-Doc QA task. For the question, we show the rank of each segment. "✔" denotes the segment is contained in the final set of QRMEM, "✘" denotes the segment is not contained in the final set of QRMEM.

D Computational Complexity Analysis

We compared various methods for handling long texts using several requests, using the GPT-3.5 model to evaluate performance on QuALITY metrics. The experiments were conducted with a window size of 4096. SLIDINGWINDOW apply a sliding window in five requests to ensure the entire text is covered. Our method achieved optimal results in comparison to others, as summarized in Table 6.

E Prompt in QRMeM

The prompt used in this paper is shown as follows, the **blue** text denotes the part to be filled with the different documents.

Table 6: Computational Complexity Comparison of Different Methods for Long Text Processing

Method	QuALITY	Number of Requests
GIST (Lee et al., 2024)	0.7005	4.32
MEMWALKER (Chen et al., 2023a)	0.6622	3.71
SLIDINGWINDOW	0.6891	5.00
QRMEM	0.7207	3.96

The prompt used for asking the LLM weather could answer the question

Read the text in triple quotes and answer a question:
"""
[Concatenated Segments]
"""
Question:
[Qeustion and possible Choice]
If the answer CANNOT be inferred from the text above, reply with action -1.
If the answer CAN be inferred from the text above, reply with action -2, and provide your reasoning and the final answer.
You are ONLY allowed to reply with action -2 or -1.
#######################
Reasoning: ...
Action: -2/-1, ...
#######################

The prompt used for Entity Recognition

Task Definition
This project aims to extract key entities from the given text to help answer related questions. Please follow the following steps when performing this task:
1. Understand the content: Before starting entity extraction, carefully read and understand the text's content. Pay attention to the story plot, themes, and context.
2. Identify key entities: Focus on important entities in the text, including but not limited to characters, locations, times, organizations, etc. These entities are usually crucial for answering questions.
3. Contextual relevance: When annotating entities, consider their roles and importance in the text. Note how entities are interconnected with events, plots, or other entities in the text. 4. Avoid subjective interpretation: Only annotate based on textual content and avoid adding personal interpretations or assumptions.
5. Review and verify: After completing the initial annotation, carefully review to ensure no omissions or errors have occurred.
The background information is
[Document Summary]
Requirement
Extract the key point. Named entities from the document below are inside the triple quote and only appear in the document.
"""
[Segment Context]
"""

The prompt used for Relation Extraction

Task Definition
This task aims to associate the extracted entities according to the text to help answer specific questions.
1. Describe the relationship in one clear paragraph;
2. Avoid false inferences: extract relationships based only on the clear information in the text, and avoid false inferences based on assumptions or external knowledge.
3. Analyze the relationships between entities: Study the mutual relationships identified by the research institute. Pay attention to how they interact and describe the relationships between entities in a concise text.
4. Record the context of the relationship: When extracting the relationship, record the specific context of the relationship, which is important for understanding the whole picture of the relationship.
Main text
"""

[Segment Content]
"""
Extracted entities
"""

[Segment Content with marked entities]
"""
Extract relations from the document according to the given entities, and give me the entity pair and the description for each relation.

The prompt used for Summary

Task Definition
Summary the context into shortened text, and retain as much relevant information as possible about the original characters, person, events, location, etc., in the summary. The context is
"""

[Segment Context]
"""
Requirement Summary the segments into a short paragraph.

The prompt used for Relation Update

Task Definition
The task is to merge the corresponding relationships based on two segments, and each segment contains a corresponding relationship. In order to merge these two relationships, we may need to pay attention to some issues and generate some corresponding questions to assist in merging these two relationships, including entities that need attention, relationships, time, temporal relations, etc.
1. Ensure entities in both segments refer to the same real-world objects or concepts and compare their attributes.
2. Verify that relationships between entities in the first segment are compatible with those in the second segment and identify any conflicts or discrepancies.
3. Check for consistent and correctly aligned time references and consider the impact of temporal relations on the merging process.
4. Refer to the document for specific guidelines or criteria for merging relationships, and review any examples or case studies that illustrate successful merging.
5. Understand how the contexts of the relationships in each segment influence the merging process and consider any external factors or conditions mentioned in the segments.
Question
[Question to be answered]
Document
[Document summary]
Relation #1
Segments
[Segment #1]
Relations
[Relations #1 to be merged]
Relation #2
Segments
[Segment #2]
Relations
[Relations #2 to be merged]