SEGMENT+: Long Text Processing with Short-Context Language Models

Wei Shi[♠], **Shuang Li**[♠], **Kerun Yu**[♡],
Jinglei Chen[♣], **Zujie Liang**[♣], **Xinhui Wu**[♣], **Yuxi Qian**[♣], **Feng Wei**[♣], **Bo Zheng**[♣],
Jiaqing Liang[♢], **Jiangjie Chen**[♠*], **Yanghua Xiao**[♠*]
[♠]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[♡]Columbia University [♣]MYbank, Ant Group
[♢]School of Data Science, Fudan University
wshi22@m.fudan.edu.cn {jjchen19, shawyh}@fudan.edu.cn

Abstract

There is a growing interest in expanding the input capacity of language models (LMs) across various domains. However, simply increasing the context window does not guarantee robust performance across diverse long-input processing tasks, such as understanding extensive documents and extracting detailed information from lengthy and noisy data. In response, we introduce SEGMENT+, a general framework that enables LMs to handle extended inputs within limited context windows efficiently. SEGMENT+ utilizes structured notes and a filtering module to manage information flow, resulting in a system that is both controllable and interpretable. Our extensive experiments across various model sizes, focusing on long-document question-answering and Needle-in-a-Haystack tasks, demonstrate the effectiveness of SEGMENT+ in improving performance.[1]

1 Introduction

Language models (LMs) have shown exceptional performance in a wide range of NLP tasks (Pu et al., 2023; Wei et al., 2022a,b). Due to the relatively short context window of most LMs, they face unique challenges in contexts such as long-document question answering, long-term memory maintenance, and processing lengthy, noisy contexts (Shaham et al., 2022; Bai et al., 2023; Packer et al., 2023; Liu et al., 2023b; Kamradt, 2023). Efficiently processing long texts across various tasks remains a core challenge in this community.

To reduce input length for handling long text, traditional retrieval is a simple and fast method but struggles with tasks requiring multiple pieces of information, often missing details and introducing noise (Wang et al., 2023). Enhancements like query rewriters (Ma et al., 2023b; Chan et al., 2024) and feedback loops (Asai et al., 2023) have reduced its usability and tied it more closely to specific downstream tasks. For handling long inputs at once, a series of works focus on broadening the context window of language models (Touvron et al., 2023; Douzon et al., 2023). This method also has intrinsic limitations: the context window cannot be infinitely expanded, and these models cannot maintain robustness across various tasks and context lengths. (He et al., 2023; Kamradt, 2023; Hsieh et al., 2024).

To access the entire text and perform multiple intermediate reasoning steps to enhance performance on complex tasks, memory management involves repeatedly using short-context LMs combined with document pre/post-processing for managing long texts (Chen et al., 2023a; Packer et al., 2023). For such systems, the key challenge lies in designing a mechanism to manage information flow across dif-

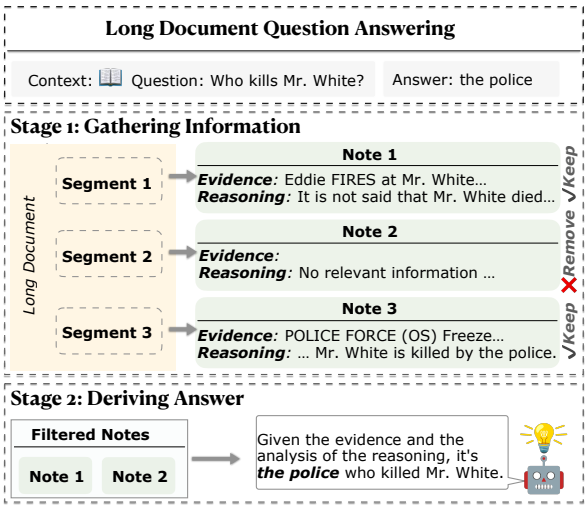

Figure 1: This picture illustrates the use of short-context models to tackle long document question answering tasks in SEGMENT+. The process begins by gathering relevant context from the document for a specific question. Only notes labeled 'keep' are used as the context to derive the final answer, avoiding noise.

*Corresponding authors.
[1]Our code is available at https://github.com/WeiShi-9/segmentplus.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 16605–16617
November 12-16, 2024

ferent invocation times. However, past research often relies heavily on the model's inherent capabilities for planning and spontaneous decision-making, resulting in an uncontrolled reasoning process with noisy, free-form text expressions.

To address the above issues and challenges, we propose the **SEGMENT+**, a robust and controllable framework that helps language models process long texts with a limited context window. The key insight of the SEGMENT+ is capturing the characteristics of queries and designing two specific components to gather and merge different types of information from long inputs. As shown in Figure 1, the SEGMENT+ agent has two stages. In stage 1, the model collects structured notes in parallel from all segments, with each note containing an Evidence and a Reasoning part. After filtering out unhelpful notes labeled as 'Remove', the remaining notes proceed to stage 2. In stage 2, the notes are divided into batches, maintaining the same order as the input, with a maximum token limit. Each batch of notes is then merged into one updated structured note. This process is iterated until the remaining notes can fit into the context window as the final context for answering the question.

The challenge of processing long texts can be effectively tackled by using two types of components for information flow control. We notice that some questions require specific detailed information, while others need further reasoning across different parts of the content. Therefore, we create an Evidence component for gathering original sentences from the input, focusing on precision, and a Reasoning component to help the model compress context into high-level semantic information, focusing on recall. This division, using both Evidence and Reasoning, makes the process both controllable and interpretable.

Significantly, retrieval and long-context language models can both benefit our method by either moderately increasing the processing context window or narrowing the input range that needs to be processed, which does not require high accuracy.

In short, our contributions include: *1)* We introduce a versatile framework for long context processing, applicable across language models of varying sizes and multiple text domains. *2)* Our method, leveraging a robust reasoning schema, outperforms other agent-based baselines and advanced long context models in long text processing tasks. *3)* We conduct a thorough analysis of SEGMENT+, highlighting the importance of structured information control.

2 Related Work

Retrieval-augmented Generation With a dense or sparse retriever, we can swiftly find relevant information in long texts by comparing query similarity. However, direct use of user queries to retrieve relevant information may not always yield useful results due to ambiguity or incomplete queries (Ma et al., 2023a; Liu et al., 2023a), thereby introducing noise (Wang et al., 2023). For re-managing the retrieved data, Zhu et al. (2023); Zhuang et al. (2023) explore the information organization capabilities of language models, utilizing LLMs as rerankers for more precise sorting. While single-turn retrieval may bring in limited useful information, insufficient for some queries, some studies (Jiang et al., 2023b; Shao et al., 2023) focus on multiple searches based on language model outputs, which may yield superior results. Despite these efforts, the information retrieved often remains fragmented, incomplete, and only partially represents the original materials from the long input. This fragmentation presents challenges for tasks requiring the synthesis and reasoning across multiple segments of a long text (BehnamGhader et al., 2023). Our method addresses this issue by using structured information gathering that includes not only the essential original evidence but also segment-aware analysis for further explanation, thus facilitating easier reasoning over the entire long input.

Long Context LMs Language models perform well in a variety of applications but struggle with large texts due to limited context windows (Shaham et al., 2022; Bai et al., 2023; Packer et al., 2023). Through techniques such as position interpolation and continuous pretraining, researchers have attempted to expand the context windows, thereby improving performance for both long and short document tasks (Chen et al., 2023b; Xiong et al., 2023). However, these approaches are limited by data quality and feasible window size constraints (Xiong et al., 2023). Besides, the models' inability to handle queries when key information is scattered across a large text is also a notable challenge (Liu et al., 2023b). He et al. (2023) discovers that this issue arises from attention failure and can be alleviated by training models with a specially designed task. Thus, the performance and robustness of models processing very long texts

in a longer context remain uncertain, and limited resources restrict our ability to indefinitely expand the model's context window. Our method addresses this challenge by dynamically optimizing the use of the available context window, thus allowing for the effective extraction of crucial information scattered throughout the text. This approach shifts the focus from merely widening the window to better utilizing the model's current context capabilities.

Memory Management Rather than processing long text in one go, we can use language models as agents to handle the long input task step by step. In such systems, memory is pivotal not only for storing out-of-window information but also as a foundation for lifelong learning through historical data analysis (Sumers et al., 2023; Majumder et al., 2023). Using LLM agents for long text tasks has its advantages. Firstly, efficiently organizing and utilizing memory can improve performance in both documents QA and dialogue tasks (Packer et al., 2023). Secondly, capturing the document structure and employing agent navigators across document segments is advantageous for document QA tasks (Chen et al., 2023a). Lastly, leveraging the task decomposition and plan-and-solve abilities of agents also benefits these tasks (Sun et al., 2023). Our approach stands out by showcasing how it's effective across variously sized models, unlike others that require high language model capabilities.

3 Method

The core challenge of processing long inputs with short-context models is how to control the information flow within different segments. In other words, how to retain the most useful information while using the fewest tokens. We first construct this process into a two-stage pipeline and then introduce each step in detail.

3.1 Problem Formulation

First, we can approach the long input processing task as a two-stage process: ***1) Gathering information*** from different parts of the input, and ***2) Reasoning over this information***, performing further inference, and eliminating duplications for the final output. From this perspective, traditional retrieval techniques directly address the first phase by selecting passages with the highest similarity to the question as context, and then explicitly carry out the second step for reasoning. Similarly, long-context-window models also perform these actions but do so implicitly to provide an answer. In contrast, our proposed SEGMENT+ clearly and efficiently performs these two actions, providing a more transparent and manageable information flow.

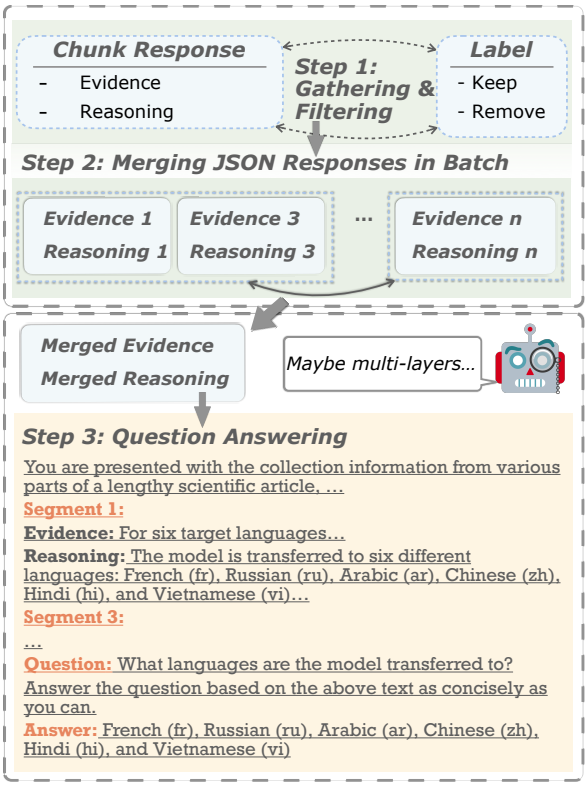

Figure 2: The proposed framework for SEGMENT+ consists of three main components. First, a gathering module collects structural information for a given query, distinguishing direct, accurate context (evidence) from the model's potentially misleading analysis (reasoning). Next, a filter module filters out noisy segments for dense information management. Finally, we merge this information in batches, taking into account the limited context window of the merging language model, to produce a suitable length context optimized for final answering.

3.2 SEGMENT+

The first question we focus on is ***how to efficiently and losslessly collect all the useful information from segments***. Before addressing this problem, we observe that common questions about long inputs can be divided into two types (Li et al., 2023; Pang et al., 2022): short-dependency questions and long-dependency questions. Short-dependency questions may only require several sentences or words for the final answer and need direct and exact information. Long-dependency questions, on the other hand, require the agent to aggregate information from different parts of the input and perform reasoning for the final answer, necessitating com-

prehensive and concisely compressed information.

Motivated by these data characteristics, we designed a specific structure named ***Note*** for gathering information that contains two components: ***Evidence*** and ***Reasoning***, which together form a set of notes. The Evidence part requires the model to collect original sentences from one segment that directly answer the asked question, corresponding to short-dependency questions and improving the precision of the information. The Reasoning part requires the model to gather possible information and compress it into concise texts related to the question, such as mentioned entities and events, corresponding to long-dependency questions and improving the recall of the information. Just like human reading habits, we may underline key sentences in the inputs and make annotations to aid further reasoning tasks.

In the first step, given a question Q and a list of potentially useful segments $C = \{c_1, c_2, \ldots, c_n\}$, for each segment c_i, we create a *Note*, which is processed in parallel. We define this process as shown in Figure 2: ***1)*** For each segment i, we collect structured information relative to the question, represented as a JSON object:

$$note_i = \{\text{“}Evidence\text{”} : \text{“”}, \text{“}Reasoning\text{”} : \text{“”}\}$$

We then filter the notes to keep them information-dense and remove noise. This is crucial for small models that lack robustness. When dealing with short-dependency questions, 10 segments produce 10 notes, but only one contains useful information. The others might state "No information" or the model might generate a possible but irrelevant answer. In such cases, the model might fail to identify the real useful information and could be influenced by crowd psychology or noise (Xie et al., 2024). In our well-structured notes, this problem can be mitigated by letting the model filter out notes with empty evidence or those that clearly state "no information" in the reasoning part. We then keep all the other notes, even if there is redundant information. As shown in Figure 2: ***1)***, each $note_i$ is labeled by a filter, denoted as F, with either a ‘Keep’ or ‘Remove’ label. We prompt language models with the question and JSON responses, evaluating their relevance based on whether they provide at least one piece of useful information. Segments that lack relevant information are tagged for removal to prevent them from leading the model toward an undesired outcome. This component can be trained using task-specific data or optimized through prompt engineering.

In the second step, the core question is ***how to best maintain the gathered information with the least tokens and without loss***. This can be addressed with a simple solution in our structured information flow control. We first keep the collected parallel notes in the same order as the segments to maintain some semantic information. Before merging these notes, we split them into batches based on a given max token limit due to the model’s context window constraints. In each batch, we concatenate the evidence parts directly, as they contain exact useful information. For the reasoning part, the model performs further inference and refines redundant information. This way, each batch yields a new merged note that maintains the structure. This process can be repeated over multiple iterations to generate a final merged note that fits within the model’s context window limit.

In the third step, to get the final answer, we use the final merged Notes as the context for model prediction. In other words, the model does not interact with the original segments.

The reasoning process in our designed framework is bottom-up. Our information collection focuses on retrieving data with a reasoning process that directly related to the final question and parts of the question or subquestions, rather than relying solely on semantic matching for information retrieval. This approach is useful for addressing both one-hop and multi-hop questions and tasks requiring knowledge synthesis from different segments. For example, in a two-hop question like *“The rapper who owns Aspiro was inspired by what when writing Song Cry?"*, the model gathers information on both the rapper’s identity and songwriting experience from each segment and performs reasoning during the merging process or final answer generation. This process can reduce costs and improve efficiency compared to multi-turn interaction searching and communications.

In conclusion, these design concepts assure a strong and efficient framework for providing accurate and comprehensive answers in the long text processing domain, including organized information collecting, strategic response labeling, and a focus on partial information gathering to solve complicated, multifaceted queries.

4 Experiments

In this section, we apply the SEGMENT$^+$ to two distinct long text reasoning tasks: long document question answering and needle-in-a-haystack tasks. For the long document QA task, we analyze SEGMENT$^+$'s ability to compress reading contexts and merge information efficiently in parallel (§4.1). For the needle-in-a-haystack task, we evaluate SEGMENT$^+$'s resistance to noise while accurately gathering essential sentences and do reasoning for final answer.(§4.2).

4.1 Long Document Question Answering

	Qasper	MSQ	HQA	NQA	QLTY
Max	19372	16337	16325	476004	8609
Min	1785	6484	1748	8961	2401
Avg	4880.54	15576.98	12793.29	75678.19	5613.78

Table 1: Summary of the maximum (Max), minimum (Min), and average (Avg) token counts for selected long document question answering datasets, tokenized using ChatGPT's tokenizer.

Understanding long documents has long been a common research issue in the NLP field, posing challenges due to the increasing length of texts and the complexities involved in comprehensive reasoning.

Benchmarks The datasets utilized in our study are extracted from two notable benchmarks in document understanding, Scrolls (Shaham et al., 2022) and Longbench (Bai et al., 2023), which are specifically designed to rigorously evaluate the capabilities of LMs in processing and reasoning through lengthy texts across diverse domains. The selected datasets include Quality (QLTY) and NarrativeQA (NQA), which focus on storytelling; Qasper, which is tailored to scientific articles; and HotpotQA (HQA) and Musique (MSQ), which are aimed at assessing factual knowledge and multi-hop question answering, akin to Wikipedia sources. The diverse range of source texts and task categories, with sample sizes of 200 for each dataset except for NQA, which comprises 100 samples, ensures a comprehensive and exhaustive evaluation of LMs across varied contexts and document lengths. For further elaboration on each dataset, readers are referred to Appendix A, while Table 1 provides the token count for each dataset.

In our evaluation process, we implement a combination of automatic metrics (Auto) and GPT-4 (OpenAI, 2023) evaluation. The GPT-4 evaluation prompt has been slightly adapted in accordance with the methodology proposed by Li et al. (2023) to change the original score range to 0-100, as detailed in the Appendix B. We mainly adopt the GPT-4 metrics because the automated metrics focus only on surface-level matching and lack semantic understanding.

Baselines We evaluate a variety of baseline models for processing long documents. Initially, we look at small-context models with a 4k token limit, followed by retrieval methods using the advanced Contriever model (Lei et al., 2023), also with a 4k token window. For broader contexts, we examined models that handle 16k tokens, which is adequate for most of our experimental data. Our review included ChatGPT (16k) (OpenAI, 2022), GPT-4 (128k) (OpenAI, 2023), Vicuna-7B (4k and 16k versions) (Chiang et al., 2023), Vicuna-13B (4k and 16k versions) (Chiang et al., 2023), and Mistral-7B-v0.2 (32k) (Jiang et al., 2023a). We keep the final stage prompt of SEGMENT$^+$ consistent with all the baselines provided by Longbench (Bai et al., 2023); more details are provided in the appendix C. The temperature for all models is set to 0 for replication purposes.

Motivated by the shared objective of leveraging limited working memory, we contrast our approach with MemGPT (Packer et al., 2023) and implement a script for automatically prompting user responses for answers. Likewise, for improved reasoning over long documents, we include Pearl (Sun et al., 2023) in our comparison. For the action mining process, we use the released resources for QULT and NQA, as they both pertain to the stories domain. We also run this process for other domains where the actions may differ.

Due to the high foundational capabilities of these methods, we tested ChatGPT and GPT-4 according to their respective settings. However, in the case of MemGPT, using ChatGPT almost never yields valid responses; the agent typically awaits user input rather than solving the question, even when task input is provided. Specifically, for the Pearl method, the model is required to generate and execute a plan, which may fail. In such instances, invalid responses are counted as errors.

Model	Method	Qasper		MSQ		HQA		NQA		QLTY	Avg.
		GPT4	Auto	GPT4	Auto	GPT4	Auto	GPT4	Auto	GPT4	GPT4
ChatGPT	MemGPT (16k)	-	-	-	-	-	-	-	-	-	-
	Pearl (16k)	51.46	31.19	26.47	9.06	38.55	12.85	20.57	5.83	64.63	40.34
	Vanilla (4k)	47.75	31.91	23.67	24.30	54.55	45.24	29.95	21.57	70.00	45.18
	Retrieve (4k)	46.80	33.36	26.40	26.27	63.12	53.56	32.50	21.28	73.50	48.46
	Long (16k)	54.10	**36.78**	**39.93**	**37.11**	**68.45**	56.30	38.45	26.78	74.50	55.09
	Segment$^+$ (4k)	**56.29**	25.66	36.38	30.81	67.96	**56.76**	**42.00**	**26.88**	**75.00**	**55.53**
GPT-4	MemGPT (128k)	55.90	22.60	39.58	33.42	67.90	50.03	48.21	19.15	74.47	57.21
	Pearl (128k)	61.03	36.01	40.13	12.25	64.77	18.47	38.38	10.00	81.92	57.25
	Vanilla (16k)	51.38	36.33	27.63	26.92	55.88	47.56	37.90	22.64	77.00	49.96
	Retrieve (4k)	51.98	35.47	34.15	32.27	70.96	59.06	50.00	**50.00**	83.50	58.12
	Long (16k)	54.72	**38.54**	**51.15**	**50.53**	79.07	**67.82**	41.50	26.31	**90.50**	63.39
	Segment$^+$ (4k)	**63.52**	25.37	48.82	44.97	**80.00**	65.79	**54.45**	30.86	88.50	**67.06**
Vicuna-7B	Vanilla (4k)	35.65	20.33	12.38	6.23	38.85	21.69	12.05	9.14	37.50	27.29
	Retrieve (4k)	39.75	**24.15**	17.48	**8.51**	**46.17**	**23.84**	**25.61**	**16.62**	33.00	32.40
	Long (16k)	30.14	21.36	13.90	7.43	43.98	22.69	20.30	12.59	40.00	29.66
	Segment$^+$ (4k)	**39.80**	14.94	**19.00**	8.19	44.42	19.55	23.85	11.52	**46.00**	**34.61**
Vicuna-13B	Vanilla (4k)	29.50	18.35	16.38	13.04	42.82	30.02	22.90	13.77	42.00	30.72
	Retrieve (4k)	37.65	**23.22**	**21.15**	**18.27**	**52.45**	**42.78**	29.60	17.97	48.50	37.87
	Long (16k)	23.53	15.75	14.28	8.79	43.60	29.26	28.65	**18.15**	51.00	32.21
	Segment$^+$ (4k)	**51.62**	17.00	16.43	11.83	42.05	31.68	**34.70**	14.75	**52.50**	**39.46**
Mistral-7B	Vanilla (4k)	50.70	21.84	17.13	11.93	43.73	25.15	18.55	12.53	55.50	37.12
	Retrieve (4k)	51.62	21.84	23.52	14.89	56.83	33.04	29.50	16.83	61.50	44.59
	Long (16k)	**59.73**	**27.09**	24.55	**17.03**	**63.58**	**35.10**	30.70	**19.38**	**65.00**	**48.71**
	Segment$^+$ (4k)	54.83	17.00	**26.98**	12.62	56.70	32.19	**37.32**	14.75	59.00	46.97

Table 2: Comparison of main results across various models and datasets. The context window in parentheses refers to the working window size limited for comparison. The highest score in each column is highlighted in **bold**. Scores are measured using the F1 metric for the 'Auto' column, while the 'GPT4' column reflects the evaluation scores of GPT-4. Segment$^+$ achieves the highest performance relative to other baselines, with the exception of Mistral-7B, which shows comparable performance in settings with the 16k-contexts model. It particularly outperforms agent-like baselines such as MemGPT and Pearl.

Main Results Our method, SEGMENT$^+$, exemplifies unparalleled adaptability across LMs of all sizes by segmenting tasks into digestible pieces. This approach empowers smaller models, such as Vicuna-7B and Mistral-7B-v0.2, to excel against a diverse array of benchmarks with remarkable efficiency.

The stronger the base model, the greater the performance gain. For larger models, our SEGMENT$^+$ significantly surpasses all comparative baselines., delivering a over 20% performance improvement over vanilla models, which highlights SEGMENT$^+$'s remarkable capability. Significantly, SEGMENT$^+$, alongside Pearl, achieves substantial performance enhancements with GPT-4 compared to ChatGPT. This underscores the fact that agent-based frameworks, when coupled with a meticulously designed reasoning schema and enhanced computational capabilities, can realize notable progress. This leap forward underscores the critical role of a systematic structure in boosting model performance, especially in the nuanced realm of long document question answering.

Nonetheless, the robustness of such a system is equally vital. MemGPT, when paired with ChatGPT, often fails to respond, terminates abruptly, or excessively depends on human input, undermining reliability. This contrasts sharply with previous agent-based methods that struggled without robust model foundations. Unlike MemGPT, which can

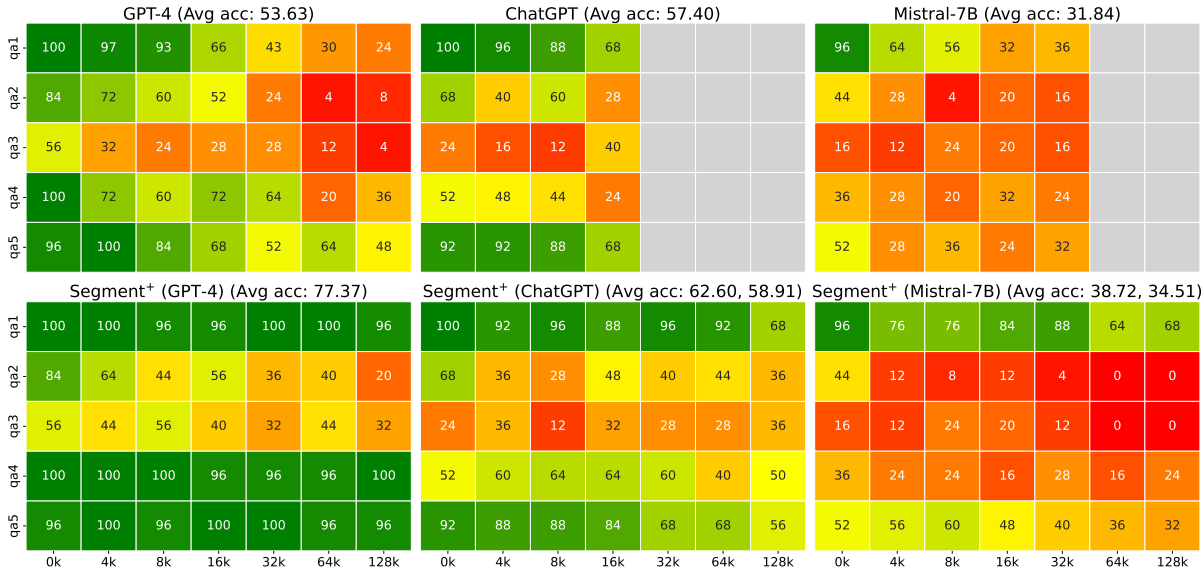

Figure 3: Babilong (Kuratov et al., 2024) Test Performance Comparison. The x-axis represents the length of the input. The y-axis shows the Exact Match (EM) performance on the Babilong task. Results for GPT-4 are taken from Babilong, with each task consisting of 25 items, consistent with the Babilong setting. The average accuracy (Avg acc) for vanilla models and SEGMENT+ (GPT-4) denotes the mean score of all colored cells. However, for SEGMENT+ (ChatGPT) and SEGMENT+ (Mistral-7B), we calculate two average scores: the initial score represents the average over valid contexts for comparison with vanilla models, while the subsequent score indicates the average over all cells. Green indicates higher performance, while red signifies lower performance. SEGMENT+ enhances overall accuracy and maintains stable performance as context length increases.

stall, awaiting further instructions, SEGMENT+'s well-designed process and schema ensure it remains effective and robust. This demonstrates SEGMENT+'s capability to navigate the complexities of LMs effectively, advocating strongly for its adoption. For smaller models, SEGMENT+ not only achieves a performance increase of over 20% over vanilla models but also surpasses the performance gains of retrieval models, further demonstrating our method's robustness.

4.2 Needle-in-a-Haystack Question Answering

Needle-in-a-Haystack (Kamradt, 2023) has recently become a popular task for testing the processing of long texts. However, we do not choose the original task because it is too artificial. We believe that the reasoning task is more suitable for evaluating long input processing. We follow Levy et al. (2024), who reported a decline in the reasoning performance of LMs as the input size increases across various tasks in similar settings.

Benchmark We adapt the Babilong benchmark (Kuratov et al., 2024), which poses a significant challenge as it requires the model to extract and process distributed facts within extensive texts, culminating in reasoning to arrive at a final answer. This tests the model's ability not only to find relevant information but also to reason over it. In line with the main experimental settings of Babilong, we select tasks from qa1 to qa5 for evaluation, using the 'Evidence' part of the collected notes for information processing. The context ranges from '0k' to '128k' tokens, where '0k' indicates a context-free environment containing only the given facts, and '4k' to '128k' denotes contexts that include these facts along with noisy data. Given that the output format is fixed, we employ an exact match approach to measure accuracy (%).

Baselines Following the experimental setting of Babilong, we have chosen GPT-4 (128k) (OpenAI, 2023), ChatGPT (16k) (OpenAI, 2022), and Mistral-7B-v0.2 (32k) (Jiang et al., 2023a) for comparison. We keep the final stage prompt of SEGMENT+ consistent with all the baselines provided by Babilong. The temperature for all models is set to 0 for replication purposes.

Results SEGMENT+ demonstrates superior performance on the Babilong tasks compared to all

models, showcasing its robust anti-noise capabilities and efficient information gathering and reasoning. Notably, with SEGMENT$^+$, ChatGPT even surpassed the performance of vanilla GPT-4, achieving a 5.28% higher accuracy. Furthermore, the stronger the base model, the greater the performance gains observed. This is likely due to the base model's enhanced capabilities, which better leverage the SEGMENT$^+$ strategy to achieve improved performance.

The performance of SEGMENT$^+$ on Babilong tasks remains relatively stable as the length of the input text increases. This stability is due to our method's ability to decompose the task's complexity during long input processing, allowing the model to process only a small piece of text at a time.

4.3 Ablations

Do filtered and structured information play crucial roles in the effectiveness of our framework? To analyze this question, we establish three ablation baselines. First, to examine the effect of information filtering, we eliminate the labeling process and rely solely on the SEGMENT$^+$, using all chunks to generate the final answer. Second, to assess the impact of structured information, we disregard the structured format and simply aggregate filtered answers from various chunks to formulate the final response in free text. Finally, to evaluate the system without both the structure and filter, we run the chunk and merge algorithm.

Results Experiment results indicate that both design elements contribute to performance. Additionally, for Vicuna-7b, the filtering module plays a more important role. Furthermore, on the Musique dataset, which tests reasoning with challenging distractors and avoids shortcuts, SEGMENT$^+$ demonstrates efficiency on complex multi-hop questions(see Appendix D). Our methodology substantially improves task performance over vanilla models due to two key factors: ***1)*** the filtration of pertinent information, and ***2)*** a structured process for integrating information. This approach not only expands the context window but also processes content more efficiently.

4.4 Segment Size Analysis

Given our use of the segmenting method, it is essential to analyze whether different segment sizes influence the performance of our approach. We examine performance variations as segment size increases from 1000 to 3000 tokens, in increments of 500. Additionally, we average the results across all five datasets in long document processing tasks.

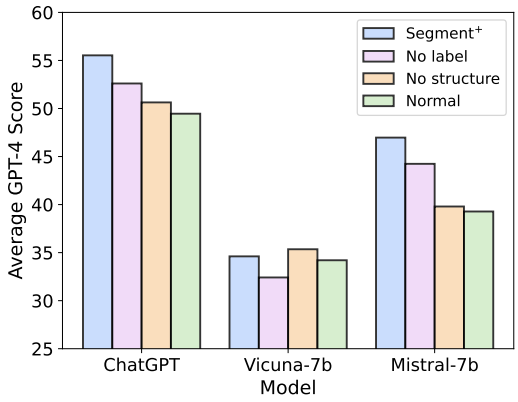

Figure 4: Ablation study results. 'No Label' refers to the condition without information filtering, 'No Structure' refers to the absence of a structured prompt, and 'Normal' indicates the model operates without both filtering and structured prompts. The results demonstrate that both design elements contribute to the final performance.

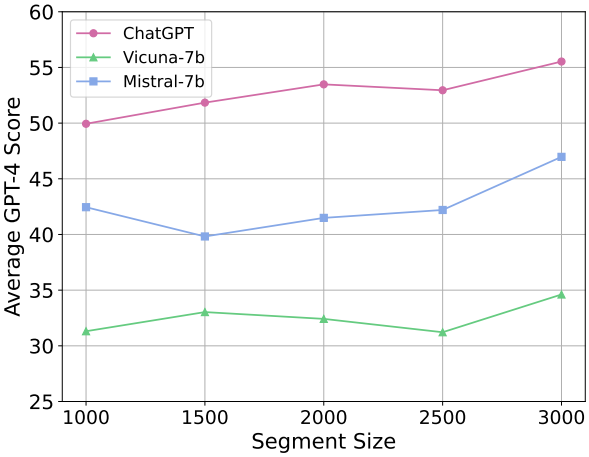

Figure 5: Segment Size Results. The average performance in long document question-answering tasks remains stable across different segment sizes, with optimal results achieved at a segment size of 3000.

Results The average performance of SEGMENT$^+$ across different segment sizes appears stable, with higher performance at the 3000 segment size. A larger segment size brings these advantages: ***1)*** The information within one segment is more complete, reducing the model's pressure to integrate information. ***2)*** Fewer segments lead to faster prediction speeds, improving efficiency.

5 Conclusion

In this paper, we introduce Segment$^+$, a simple yet effective plug-and-play methodology designed to augment the processing of long inputs within limited context windows, leveraging structured information flow control motivated by data characteristics and filtering mechanisms. Our extensive experiments and analyses substantiate that Segment$^+$ significantly enhances performance in long document question answering and noisy text processing, thereby illustrating its broad applicability across diverse domains in this field. When compared to agent-based methods, Segment$^+$ not only achieves superior performance but also exhibits greater stability. Furthermore, this information control schema holds potential for broader applications in scenarios requiring long input processing, such as in agent memory management and video information processing.

Limitations

First, our method is primarily focused on document input processing; it cannot be directly applied to more complex structured texts such as code or text-based games. However, we believe that the underlying concept can be adapted to design specific structures for these scenarios. Second, we notice that SEGMENT$^+$ is more effective when applied to stronger models. This may be due to the strong models' good instruction-following abilities, allowing them to adhere well to our schema design, and their robustness, providing better resistance to textual noise and enhanced reasoning capabilities.

Ethics Statement

This paper introduces a novel framework for long-context processing, evaluated on publicly available datasets such as Scrolls, Longbench, and Babilong; therefore, no specific ethical considerations are addressed.

Acknowledgement

We are deeply grateful to Yikai Zhang, Jian Xie, and Siyu Yuan from Fudan University for their insightful suggestions and thoughtful discussions, which greatly contributed to this work. This work was supported by Ant Group Research Fund.

References

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

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, multitask benchmark for long context understanding.

Parishad BehnamGhader, Santiago Miret, and Siva Reddy. 2023. Can retriever-augmented language models reason? the blame game between the retriever and the language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15492–15509, Singapore. Association for Computational Linguistics.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation.

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.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Thibault Douzon, Stefan Duffner, Christophe Garcia, and Jérémy Espinas. 2023. *Long-Range Transformer Architectures for Document Understanding*, page 47–64. Springer Nature Switzerland.

Junqing He, Kunhao Pan, Xiaoqun Dong, Zhuoyang Song, Yibo Liu, Yuxin Liang, Hao Wang, Qianguo Sun, Songxin Zhang, Zejian Xie, and Jiaxing Zhang. 2023. Never lost in the middle: Improving large language models via attention strengthening question answering.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models?

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. 2023a. Mistral 7b.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation.

G. Kamradt. 2023. Llmtest: Needle in a haystack. https://github.com/gkamradt/LLMTest_NeedleInAHaystack.

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

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. In search of needles in a 11m haystack: Recurrent memory finds what llms miss.

Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. Unsupervised dense retrieval with relevance-aware contrastive pretraining. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10932–10940, Toronto, Canada. Association for Computational Linguistics.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2023. Loogle: Can long-context language models understand long contexts?

Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and Ji-Rong Wen. 2023a. Reta-llm: A retrieval-augmented large language model toolkit.

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

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023a. Query rewriting for retrieval-augmented large language models.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023b. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. 2023. Clin: A continually learning language agent for rapid task adaptation and generalization.

OpenAI. 2022. Chatgpt powered by gpt-4. https://www.openai.com/chatgpt.

OpenAI. 2023. Gpt-4 technical report.

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

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

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy.

Theodore R Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L Griffiths. 2023. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*.

Simeng Sun, Yang Liu, Shuohang Wang, Chenguang Zhu, and Mohit Iyyer. 2023. Pearl: Prompting large language models to plan and execute actions over long documents.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. 2023. Effective long-context scaling of foundation models.

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.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. Open-source large language models are strong zero-shot query likelihood models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8807–8817, Singapore. Association for Computational Linguistics.

A Dataset Description

Datasets from the validation set of Scolls (Shaham et al., 2022):

- **QUALITY (Pang et al., 2022)**: Deep textual engagement is required for this concise yet difficult multiple-choice question set about stories and movies. There is one correct answer for each question, which is meant to assess thorough comprehension rather than just skimming or reading an overview.
- **NarrativeQA (Kočiský et al., 2018)**: The movie scripts that make up this dataset were gathered from different websites. It requires models to produce free-text responses to predetermined queries. The task pushes the models to go beyond simple pattern matching or salience cues, forcing them to engage in deep reasoning over lengthy scripts or books.
- **Qasper (Dasigi et al., 2021)**: This dataset, which is distinguished by its logical and well-structured content, focuses on providing answers to questions within the context of academic research papers. Because the question collectors were not exposed to the entire content of the papers, some questions might go into great detail or even be unanswerable.

Datasets from the test set of Longbench (Bai et al., 2023):

- **HotpotQA (Yang et al., 2018)**: The dataset under focus is a question-answering set derived from Wikipedia that requires multi-hop reasoning over various passage segments.
- **Musique (Trivedi et al., 2022)**: With an emphasis on minimizing train-test leakage, this multi-hop reasoning dataset is designed to get around the shortcuts found in datasets of a similar nature. In order to increase the difficulty and put a model's reasoning skills to the test, it also presents increasingly complicated distractor contexts.

B GPT-4 Evaluation Prompt

The evaluation prompt of the Qasper, HQA, MSQ, and NQA datasets was conducted using a prompt structure based on the methodology described in (Li et al., 2023).

> There is a ground truth answer to a question and an auto-generated answer. Please compare the generated answer with the ground truth and evaluate the generated answer from the perspectives of information completeness, consistency, fluency, and grammar by giving a score within the range of 0 to 100.
> Question = *question*
> Groundtruth answer = *answer*
> Generated answer = *prediction*
> Score =

The evaluation prompt of the Quality dataset is shown below.

> Give you a 4-choice question and its correct answer (only one choice is correct). You need to check whether the model prediction answer is correct or not. Let's do it step by step.
> 1. You should carefully read the first and the last sentence of the model prediction. If more than one choice is mentioned in the prediction, you should read the whole prediction carefully and figure out the final predicted answer.
> 2. Turn the answer into (A), (B), (C) or (D).
> 3. If the correct answer is choosed and is the only choosed answer, then you can say 'true'. If the model give false, none or multi-answers, you should give 'false'.
> Question: *question*
> Correct answer: *answer*
> Model prediction: *prediction*
> Model predicted options:
> Correct option:
> Evaluation:

C Long Document Question Answering Prompt

The query prompt for HotpotQA, which is slightly modified for other datasets to suit task descriptions.

> You are provided with a segment from a long document along with a question related to this document.
> Segment Content: segment
> —
> Question: question
> —
> Your task: Evaluate the provided segment against the question to identify and categorize information into two distinct types: "Evidence" and "Reasoning". Your assessment and

categorization should adhere to the following guidelines:

Guidelines for Note-Writing:

Your note should be meticulously structured into two main parts: Evidence and Reasoning, following these guidelines:

- Evidence:

1. Extract key sentences or descriptions from the segment that are pertinent to the question, with a focus on specific details such as numbers, relevant words, and other significant elements.

(1) Include content that directly relates to the question, providing a straightforward answer.

(2) Also include content that may not directly answer the question but is valuable for answering it when combined with information from other segments. For instance, for questions about someone's birthplace, include all mentioned birthplaces for potential matching in later analysis. Similarly, if the question involves several events but this segment only contains information about one event, you should include it.

2. Accurately quote the directly related sentences to present clear and unambiguous evidence.

- Reasoning:

1. Analyze the question and any subquestions, offering answers, summaries, interpretations, or any relevant commentary to deepen the understanding of the question.

The note should be formatted in JSON as follows:

{ "Evidence": "Your evidence content here",

"Reasoning": "Your reasoning content here"}

The merge prompt for HotpotQA, which is the same for other datasets to suit task descriptions.

You are presented with the collection of information from various parts of a lengthy document, along with a specific query that requires a response. The collected information is clearly divided into two parts: Evidence and Reasoning. The Evidence comes from original content of the article, the Reasoning is the model's interpretation based on this evidence.

Collected information: notes

—

Question: question

—

Detailed Instructions: Process the information from these notes in two separate parts: merge the Evidence sections together and then merge the Reasoning sections.

1. Evidence Synthesis: Examine the Evidence section closely, preserving original content that could possibly help answer the query. Aim to retain as much information as possible without omission.

2. Reasoning Enhancement: Ensure the reasoning is clear, well-structured, and concisely addresses the query within 1-2 sentences.

Upon completing your analysis, update the collected information in the following JSON format:

{ "Evidence": "Your evidence content here",

"Reasoning": "Your reasoning content here"

}

D Ablation Study on Musique Dataset

Model	Segment+	No Label	No Structure	Normal
ChatGPT	67.96	35.325	33.65	28.55
Vicuna-7b	44.42	14.35	26.20	20.70
Mistral-7b	52.35	24.02	22.52	16.38

Table 3: Evaluation of ablation baselines on the Musique dataset using GPT-4 scores.