Enhancing Large Language Model with Self-Controlled Memory Framework

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are constrained by their inability to process lengthy inputs, resulting in the loss of critical historical information. To address this limitation, in this paper, we propose the Self-Controlled Memory (SCM) framework to enhance the ability of LLMs to maintain long-term memory and recall relevant information. Our SCM framework comprises three key components: *an LLM-based agent* serving as the backbone of the framework, *a memory stream* storing agent memories, and *a memory controller* updating memories and determining when and how to utilize memories from memory stream. Additionally, the proposed SCM is able to process ultra-long texts without any modification or fine-tuning, which can integrate with any instruction following LLMs in a plug-and-play paradigm. Furthermore, we annotate a dataset to evaluate the effectiveness of SCM for handling lengthy inputs. The annotated dataset covers three tasks: long-term dialogues, book summarization, and meeting summarization. Experimental results demonstrate that our method achieves better retrieval recall and generates more informative responses compared to competitive baselines in long-term dialogues. [1]

1 Introduction

Recently, Large Language Models (LLMs) have attracted significant attention due to their remarkable performance in various tasks (Brown et al., 2020a; Zeng et al., 2023; Ouyang et al., 2022; Thoppilan et al., 2022). Instruction-tuning (Raffel et al., 2020; Wei et al., 2022a; Chung et al., 2022) helps LLMs comprehend natural language task descriptions, while reinforcement learning with human feedback (Schulman et al., 2017; Stiennon et al., 2020; Bai et al., 2022) aligns generated text with human preferences.

[1] https://anonymous.4open.science/r/SCM4LLMs-ABA2

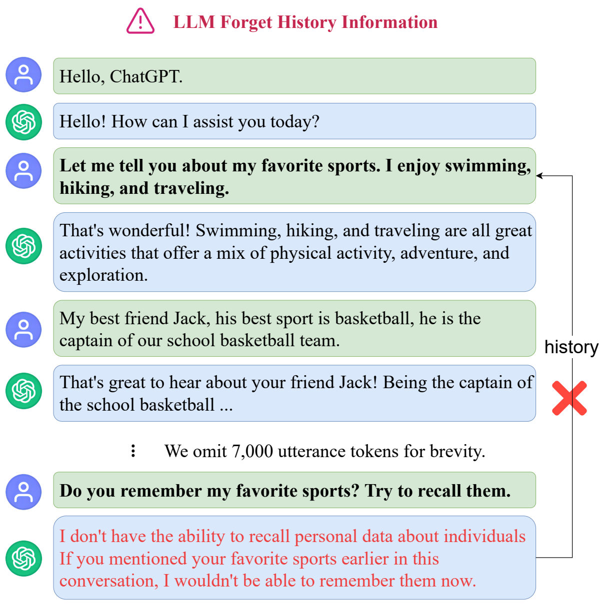

Figure 1: An example of LLM forgetting historical information. In the long-term dialogue, when the user mentions a hobby-related topic discussed in a previous conversation, ChatGPT forgets the information due to excessive historical noise.

LLMs offer numerous advantages, but their utility is hindered by two main factors: the maximum input length and the computational complexity of self-attention (Wang et al., 2020; Press et al., 2022). Although some models (OpenAI, 2022) are capable of processing long inputs, they may still struggle to capture crucial contextual information in exceptionally lengthy texts. As illustrated in Figure 1, even ChatGPT can overlook crucial contextual information from preceding text due to the accumulation of historical noise.

To address this limitation, we propose the **S**elf-**C**ontrolled **M**emory (SCM) framework, enabling LLMs to process text of infinite length without the need for any modifications or additional training. Our SCM framework consists of three essential components: *an LLM-based agent* that serves as the core component, *a memory stream* that stores the agent's memories, and *a memory controller* responsible for updating the memories and determining when and how to utilize them from the memory stream. In this framework, the input text is divided into segments, which are then provided to the LLM as observations (inputs). Each segment is processed by the LLM using two types of memory: a long-term memory (activation memory) that retains historical information and a short-term memory (flash memory) that captures real-time memory information from the preceding segment. During each processing step, the memory controller makes decisions to introduce only necessary memory information to avoid introducing additional noise.

Furthermore, we annotate a dataset to evaluate the effectiveness of SCM for handling lengthy inputs. The annotated dataset covers three tasks: long-term dialogues, book summarization, and meeting summarization. Notably, the number of tokens per instance ranges from 20 thousand to 2 million surpassing the capabilities of conventional large language models with context windows smaller than 4k, which are ill-equipped to handle such extensive textual input. Our experimental results demonstrate that the integration of the SCM framework with text-davinci-003 (non-dialogue-optimized LLM) effectively outperforms ChatGPT and surpasses strong baseline models when confronted with ultra-long inputs or long-term dialogues. For summarization tasks, our SCM-based approaches exhibits significantly superior performance in terms of coherence and coverage in generating summaries compared with baseline model.

In this paper, we summarize the key contributions as follows:

- We propose the Self-Controlled Memory (SCM) framework to unleash infinite-length input capacity for LLMs, which can decide when and how to introduce memory information to generate the response.
- We contribute a dataset to evaluate the effectiveness of SCM in three tasks: long-term dialogues, book summarization, and meeting summarization.
- Our proposed SCM framework does not require any modification or fine-tuning of LLMs, making it highly scalable in terms of memory stream.

2 Self-Controlled Memory

Here, we provide a detailed description of our proposed the self-controlled memory (SCM) framework, as illustrated in Figure 2. Firstly, the workflow of SCM will be briefly introduced in Section 2.1. Subsequently, the three key components of SCM will be presented: (1) an LLM-based agent (Section 2.2) serving as the backbone of the framework, (2) a memory stream (Section 2.3) storing agent memories, and (3) a memory controller (Section 2.4) updating memories and determining when and how to utilize memories from memory stream.

2.1 Workflow of SCM

As illustrated in Figure 2, the workflow of SCM consists of six explicit steps. Initially, the agent acquires observation at turn T. Following this, the memory activation process begins, where the memory controller determines if it is necessary to activate memory based on the current observation. Next, memory retrieval is initiated, using the observation as a query to retrieve top K-ranked memories. The fourth step involves memory reorganization, wherein the controller decides whether to use the original or summarized memory directly. Subsequently, the framework combines the retrieved memories in a predefined format, providing background information for response generation. The fifth step, input fusion, involves the predefined prompt that fuses the restructured memory with the present observation, serving as the model's input. The details of this prompt are shown in Figure 7. Lastly, the LLM-based agent generates a response based on the previous step's result, incorporating the current interaction, including observation and response, into the memory stream.

2.2 LLM-based Agent

The LLM-based agent serves as the core component of our SCM framework by generating coherent and accurate responses based on well-designed instructions (e.g., in Figure 3 and Figure 5). In this work, we adopt two powerful LLMs, *text-davinci-003* and *gpt-3.5-turbo*, as agents in our SCM framework, respectively.

2.3 Memory Stream

The memory stream stores all historical memory items and can easily achieve high-speed access through cache storage technologies such as Redis

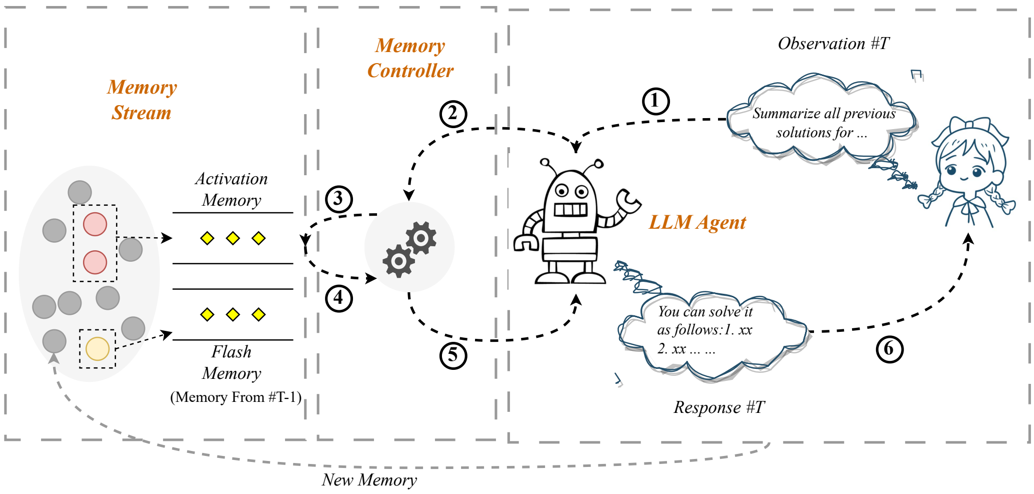

Figure 2: The workflow of our proposed Self-Controlled Memory(SCM) framework, where numbers 1-6 represent the six explicit steps of one iteration with new observation #T. These steps are (1) Input Acquisition; (2) Memory Activation; (3) Memory Retrieval; (4) Memory Reorganization; (5) Input Fusion; (6) Response Generation.

or vector databases like Pinecone [2]. Specifically, each memory item consists of (1) an interaction index, (2) an observation, (3) a system response, (4) a memory summarization (refer to the next paragraph for elaboration) and (5) an interaction embedding that illustrates the current interaction semantics. To obtain the interaction representative embedding, we combine the textual content of both the observation and system response and utilize the *text-embedding-ada-002* model [3] to get the embedding vector of the text. When memory retrieval is necessary, the memory stream retrieves and returns two kinds of items: Activation Memory, which stores related historical memories, and Flash Memory, which stores interaction memories of the previous turn $T-1$.

Memory Summarization Memory summarization plays a vital role in processing lengthy inputs, where a single interaction or dialogue turn can consist of more than 3,000 tokens. Obtaining the key information of individual turns through turn summarization is a non-trivial task when attempting to integrate multi-turn information within a limited contextual window. Figure 3 shows the English prompt that is specifically designed for memory summarization in individual interactions (i.e., dialogue tasks). In addition, other language versions of the prompt can be found in Appendix A.

Memory Retrieval In our study, we employ an empirical approach of concatenating the observation summary and system response summary (i.e., the memory summarization result of each item)

[2]https://www.pinecone.io/
[3]openai-text-embedding document

> Below is a conversation between a user and an AI assistant. Please provide a summary of the user's question and the assistant's response in one sentence each, with separate paragraphs, while preserving key information as much as possible.
>
> Conversation:
>
> User: {user input}
> Assistant: {system response}
>
> Summary:

Figure 3: Prompt for dialogue memory summarization.

to derive semantic representations for individual items. This concatenation is necessary due to the potential significant variation in length between the observation and system response within the memory stream. Such variation can create an imbalance in the semantic information captured solely from the original texts. Consequently, directly utilizing semantic vectors obtained from the original texts may not effectively balance the semantic information between observations and system responses.

2.4 Memory Controller

This section focuses on the central component: the memory controller, and its workflow is illustrated in Figure 4. The primary objective behind the design of the memory controller is to introduce the minimum necessary information to avoid excessive noise that may disrupt the model's performance.

Specifically, this can be divided into three scenarios for discussion. Firstly, not all observations,

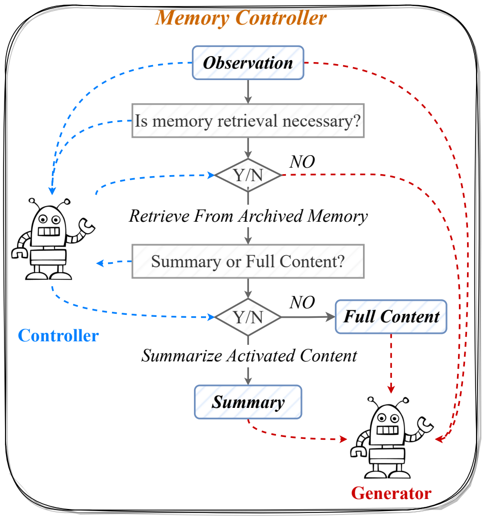

Figure 4: Workflow of the Memory Controller.

also referred to as user input or instruction, require access to historical memory. For instance, the user instruction "Tell me a joke" does not necessitate retrieving the user's historical memory. However, certain user input, such as "Do you remember the conclusion we made last week on the fitness diets" requires the retrieval of past memories.

Secondly, the amount of memory can be enormous, ranging from hundreds to thousands or even tens of thousands. A controller is needed to retrieve and filter the memory.

Thirdly, given the limited input length of the model, it becomes necessary for the controller to determine whether to employ the full content of the memory or a summary of it. The original full text can be excessively long and may exceed the model's maximum length capacity.

In the subsequent subsections, we present the detailed workflow of the memory controller, which considers each of the aforementioned scenarios.

Memory Controller Workflow As illustrated in Figure 4, the memory controller is designed to determine when to retrieve memories and how to utilize the retrieved memories in response to a novel observation.

The controller is also a language model, which controls the entire process by self-asking two questions:

1. Is it necessary to activate memories given current user input?

2. Can the current user input be answered correctly using only the summary of memory?

Given a user command, determine whether executing the command requires historical or previous information, or whether it requires recalling the conversation content. Simply answer yes (A) or no (B) without explaining the information:

User Command: {User Input}

Answer:

Figure 5: English prompt for the necessity of using memory.

Activate Memories To address the first question, we have devised a prompt for the controller to determine whether or not to activate memories. This prompt is illustrated in Figure 5. If the model responds with "yes(A)", relevant memories will be activated to provide an answer to the current question. During the process of retrieving memories, we employ the current observation as a query and assess the rank score of each memory based on two factors: **recency** and **relevance**. The recency factor places high importance on memory items that have been accessed recently, emphasizing the agent's attention on the most recent interactions. Furthermore, the relevance score of each memory is computed by calculating the cosine similarity between the current query embedding and the memory embedding.

The final rank score of each memory is determined by summing its recency and relevance scores: $rank\_score = recency\_score + relevance\_score$. Depending on the length limit, we select the top k memories with the highest rank scores as the activated memories. Here, the value of k can range from 3 to 10.

Use Summary To address the second question, we have designed a prompt to evaluate whether the user's question can be answered using the turn summary. This prompt is depicted in Figure 6. We perform this evaluation for each activated memory that exceeds 800 tokens. It is important to highlight that the summary assessment takes place only when the total number of activation memory tokens surpasses 2000. If the assessment yields a positive result, indicating that the summary can indeed answer the user's question, we utilize the memory summary to represent that specific memory.

```
Given the conversation content and the user
question, please answer the command question.

Conversation Content: ```{content}```
User Question: ```{query}```

Command Question: Based on the conversation
content, can the user question be answered by
conversation content? Respond with (A) for yes,
(B) for no.

Please strictly follow the format below to answer
the questions:
[Answer]: (A) / (B).
```

Figure 6: English prompt for whether or not to use the summary of memory.

```
Here is a conversation between a user and an AI
assistant. Please answer the user's current
question based on the history of the conversation:

History of the conversation: {history_turn}

Previous conversation: {last_turn}

###

User: {user_input}
Assistant:
```

Figure 7: English Prompt of ultra-long dialogue generation.

3 Experiments

To evaluate the effectiveness and robustness of the SCM framework, we conduct extensive experiments on three tasks, long-term dialogues, book summarization, and meeting summarization. Then, we investigate whether memory-enhanced LLMs can offer more comprehensive coverage and create coherent contextual logic summaries compared to traditional LLMs when tackling long text summarization scenarios.

3.1 Evaluation Benchmark

To evaluate the SCM performance across various scenarios, we collect open-source data from ShareChat[4], online book websites[5], and the VC-SUM dataset (Wu et al., 2023). Then, we utilize human annotation to create probing questions and summaries for the collected data. The dataset statistics are illustrated in Table 1.

[4] https://paratranz.cn/projects/6725
[5] https://www.gutenberg.org/

	Dialogue	Book	Meeting
#Instances	18	10	20
Max tokens	34k	2M	50k
Total tokens	420k	8M	632k
Max turn	200	-	80
Language	En+Zh	En+Zh	Zh

Table 1: Evaluation dataset statistics. 2M means 2 million token count.

3.2 Baselines

To ensure a fair comparison, we have selected specific model variants for experimental analysis: (1) SCM$_{\text{turbo}}$: Utilizing *gpt-3.5-turbo-0301* as the backbone of our SCM framework. (2) SCM$_{\text{davinci003}}$: Utilizing *text-davinci-003* as the backbone for SCM framework. (3) SCM$_{\text{davinci003}}$ *w/o* memory controller: Remove the memory controller and concatenate the full retrieved content. If the token length of the concatenated history exceeds 2500, truncate it. (4) SCM$_{\text{davinci003}}$ *w/o* flash memory: Remove the flash memory (short-term memory), which contains the latest information. (5) SCM$_{\text{davinci003}}$ *w/o* activation memory: Remove the activation memory (long-term memory), which is essential for answering questions involving long-distance dependencies.

3.3 Main Results

To quantitatively compare the performance of the models, 105 test questions are annotated based on the dialogue data and categorize them into two groups: single-turn related questions and multi-turn related questions. Additionally, for evaluating the two summarization tasks, we compare the performance of SCM variants with the baseline model.

Evaluation Metrics Distinct evaluation metrics are utilized for long-term dialogue scenario and two summarization scenario. For long-term dialogue scenario, the performance of our framework is assessed based on the following metrics. (1) Answer Accuracy: Evaluates the accuracy of answers to probing questions. (2) Memory Retrieval Recall: Determines if related memory can be successfully retrieved by memory controller. (3) Single Turn Accuracy: Examines the accuracy of answers to probing questions related to individual turns in the conversation history. (4) Multi Turn Accuracy:

Model Name	Answer Acc.	Memory Retrieval Recall	Single Turn Acc.	Multi Turn Acc.
SCM $_{\text{turbo}}$	68.3	93.5	73.5	64.3
SCM $_{\text{davinci003}}$	**77.1**	**94.0**	**79.6**	**75.0**
w/o memory controller	59.3 (-17.8)	93.8 (-0.2)	71.7 (-7.9)	49.4 (-25.6)
w/o flash memory	72.9 (-4.2)	93.9 (-0.1)	74.6 (-5.0)	74.8 (-0.2)
w/o activation memory	10.5 **(-66.6)**	0.0 **(-94.0)**	18.2 **(-61.4)**	0.0 **(-75.0)**

Table 2: Long-term dialogue evaluation results. The total number of probing questions is 105, including Chinese and English, with 49 single-turn and 56 multi-turn related questions. The lower part of the table is the ablation experiment of our framework.

Similar to single-turn accuracy, but it requires considering the multi-turn history in order to answer these probing questions. Additionally, two metrics, coverage and coherence, are used to evaluate content coverage and plot coherence in summarization tasks. To facilitate a comprehensive comparison, we assess the effectiveness of the model by comparing its win rate to that of the baseline model, namely RecursiveSum (Wu et al., 2021) by OpenAI, which first summarizes small sections of the book and then recursively summarizes these summaries to produce a summary of the entire book.

Dialogue Results Table 2 displays the long-term dialogue results and demonstrates that the SCM $_{\text{davinci003}}$ is superior to the SCM $_{\text{turbo}}$ for this particular task. This may be attributed to the SCM $_{\text{turbo}}$'s conservative nature, which can lead to hesitation in answering privacy related probing questions. In contrast, the SCM $_{\text{davinci003}}$ is capable of providing quicker and more precise responses. Moreover, we conducted an ablation study to investigate the independent effect of each module in SCM framework, the results are illustrated in the lower part of Table 2. When the activation memory is removed, the accuracy of the framework's responses experiences a significant drop, resulting in an approximate 60% decrease in performance. This is because the majority of probing questions are derived from long-distance dialogue records, which rely on activation memory to retrieve them. What's more, in the absence of activation memory, both memory retrieval recall and multi-turn accuracy have decreased to zero. This further demonstrates the significance of activation memory. However, when flash memory is removed, the performance only experienced a slight drop. This is because flash memory provides fewer clues to answer probing questions, resulting in a minor impact on the final accuracy. Removing the memory controller leads to a greater drop in accuracy for multi-turn related questions compared to single-turn questions. This is because the absence of the memory controller's dynamic memory filtering and use of summaries for efficient input token management results in the concatenation and truncation of all retrieved memories, leading to significant information loss.

Summarization Results A side-by-side comparison is performed by human annotators to check the summarization ability of our framework. The annotators have to choose which one is better based on the answers. They are blind to models and other information. Figure 8 illustrates the book and meeting summarization results. Based on the experimental results, we have obtained three conclusions: (1) SCM $_{\text{davinci003}}$ provides better coverage than SCM $_{\text{turbo}}$. (2) SCM $_{\text{davinci003}}$ and SCM $_{\text{turbo}}$ demonstrate comparable coherence performance due to their memory-enhanced mechanism. (3) The SCM framework without memory loses contextual dependency and consequently produces unsatisfactory summarization outcomes. It is evident from the model comparison results that SCM $_{\text{davinci003}}$ consistently outperforms SCM $_{\text{turbo}}$ summarizing both books and meetings. This can be attributed to the fact that SCM $_{\text{turbo}}$'s summarization primarily focuses on general principles, whereas it overlooks detailed core plots. In terms of human evaluation, the SCM $_{\text{davinci003}}$ model's results are more favored because of their conciseness, clarity, and richer plot content.

3.4 Further Analysis

The purpose of this qualitative study is to answer three research questions (RQs). The following experiment evaluates the performance of the SCM $_{\text{davinci003}}$ model without dialogue optimization in comparison to the vanilla ChatGPT model.

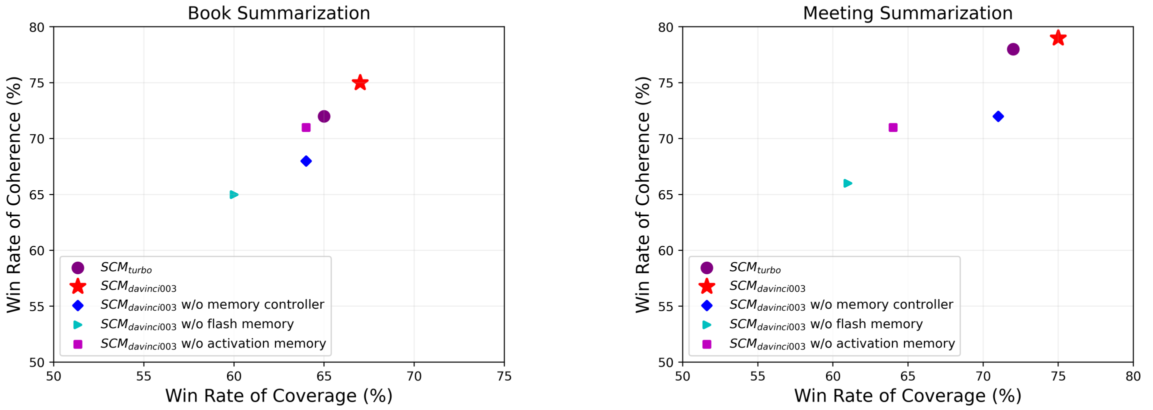

Figure 8: The win rate of SCM variants against baseline model, RecursiveSum (Wu et al., 2021) by OpenAI, in both book and meeting summarization tasks. The figure also shows a comparison of the results of the SCM framework and its various component ablations.

Figure 9: Long-term dialogue example. To answer users' questions, our model can accurately retrieve relevant memories from massive memories and generate accurate responses based on these memories.

RQ1. Can SCM framework compete with or even outperform ChatGPT within a specific token limit? **Yes**.

The example in Figure 1 includes 4k tokens, wherein the user inquired about their hobbies, discussed 100+ turns ago with the agent. The SCM framework provides an accurate response to the query, demonstrating exceptional memory-enhanced capabilities, as apparent from the observation. In contrast, it appears that ChatGPT is distracted by a considerable amount of irrelevant historical noise.

RQ2. Can SCM framework scale to provide accurate responses to users' questions, which are related to historical contexts that date back hundreds or even thousands of turns? **Yes**.

The example presented in Figure 9 illustrates a long-term dialogue comprising over 100 turns. At the outset, the user states that his goal is to reduce weight and intends to initiate a running regime. Subsequently, the user and the model converse daily about progress towards achieving their weight loss goals, among other conversation topics. After over 100 rounds of dialogue, the token length of the conversation has already exceeded 10k tokens. The user then asks the model "Do you remember my first sport?". Our SCM framework recalls sports-related information from memory and combines it with the user's current question. Afterwards, the framework generates an accurate response.

RQ3. Can SCM demonstrate effective generalization to other lengthy input scenarios?**Yes**.

Figure 10 illustrates an example of summarizing lengthy books and meetings with our SCM framework in iterative and hierarchical manner. This lengthy document has been divided into several parts and gradually summarized to obtain the first-level local summary, and then hierarchically summarized to obtain the final summary. In order to maintain context coherence, relevant memories from previous sections will be added to the input text. The conventional method involves dividing lengthy texts into separate smaller text blocks that can be processed by the model. and summarizing each text block independently. However, this method can lose the dependency relationship between paragraphs. Our SCM framework facilitates the summarization process by utilizing the related memories, thus establishing substantial coherence between the two summaries. Ultimately, the framework incorporates a divide-and-conquer strategy to

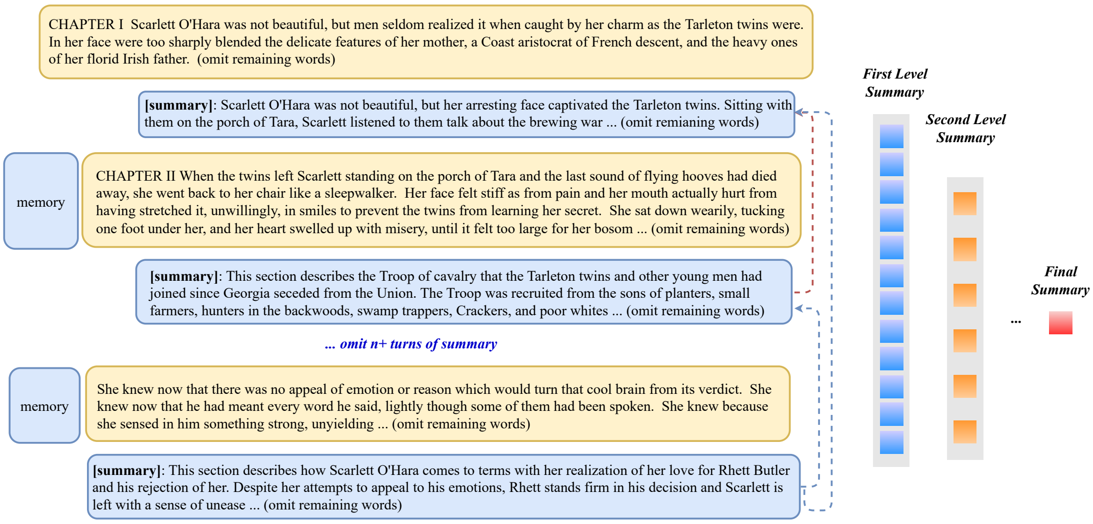

Figure 10: Ultra-long book iterative and hierarchical summarization example from *Gone With The Wind*. Our framework divides the text into small blocks and sequentially summarizes each block. We then hierarchically summarize the first level summary until reaching the final summary.

generate the final document summary. The final summary provides a comprehensive summary by utilizing information from each document block.

4 Related Work

Large Language Models Large Language Models (LLMs) are language models trained on massive amounts of text data (Vaswani et al., 2017; Devlin et al., 2019; Liang et al., 2023; Yang et al., 2020, 2021) based on the Transformer architecture. The pre-training and fine-tuning paradigm has contributed to a number of downstream language understanding and generation tasks. Subsequently, GPT-1 (Radford et al., 2018), GPT-2 (Radford et al., 2019), and GPT-3 (Brown et al., 2020b) are developed with gradually increasing parameter sizes (GPT-3 has 175B parameters). LLMs enhanced by instruction tuning have shown emergent abilities in complex reasoning (Wei et al., 2022b,c; Chai et al., 2024), knocking both academia and industry.

LLMs have achieved remarkable performance and pushed the boundaries of NLP tasks, including LAMBDA (Thoppilan et al., 2022), PaLM (Chowdhery et al., 2022), OPT (Zhang et al., 2022a), LLaMA (Touvron et al., 2023), BLOOM (Workshop et al., 2023), and Qwen (Bai et al., 2023c). But current LLMs still face severe limitations when processing tasks involving extremely long inputs.

Long Text Sequence Processing Handling long text sequences has been a persistent challenge in NLP tasks (Bai et al., 2023b; Wang et al., 2023; Pi et al., 2022; Yang et al., 2023; Bai et al., 2023a). Existing solutions mainly involve modifying the attention structure to reduce computational costs and expanding the pre-training sequence length (Beltagy et al., 2020; Zaheer et al., 2021; Guo et al., 2022; Phang et al., 2022; Dong et al., 2023). Another alternative approach (Press et al., 2022) uses special positional encoding during pre-training to enable the model to learn relative positions and handle longer input texts during inference, where the generalizability of these methods remains uncertain. In the field of long-text summarization, hierarchical or iterative methods (Wu et al., 2021; Zhang et al., 2022b; Cao and Wang, 2022; Liang et al., 2022; Zhong et al., 2023) are used to handle long texts by decomposing a complex problem into multiple sub-problems. However, these methods fail to capture the relationships among sub-problems.

5 Conclusion

In this paper, we propose a Self-Controlled Memory (SCM) framework to extend the input length of any LLMs to an unlimited length and effectively capture useful information from all historical information. This method does not require any training or modification of models. In addition, we annotate an evaluation dataset comprising three tasks. Experimental results demonstrate that SCM allows LLMs, which are not optimized for multi-turn dialogue, to attain comparable multi-turn dialogue capabilities to ChatGPT, and outperform ChatGPT in long document summarization tasks.

Limitations

One limitation of this study is that while the SCM framework has the capability to handle infinite rounds of dialogue, we evaluate its performance only in a limited setting, with a maximum of 200 dialogue turns and a 34,000 max token count of dialogue. The reason is that both qualitative and quantitative evaluations of very long texts are exceedingly difficult. Another limitation is that the SCM framework needs powerful and instruction-following LLMs like *text-davinci-003* and *gpt-3.5-turbo-0301*. However, this can be resolved when more powerful smaller LLMs are developed.

Ethical Considerations

The dataset used for evaluation in this paper is obtained from open data sources and has been manually verified and screened to eliminate any data with ethical risks and sensitive content. This ensures that the content is compliant with existing regulations and laws.

References

Jiaqi Bai, Hongcheng Guo, Jiaheng Liu, Jian Yang, Xinnian Liang, Zhao Yan, and Zhoujun Li. 2023a. Griprank: Bridging the gap between retrieval and generation via the generative knowledge improved passage ranking.

Jiaqi Bai, Ze Yang, Jian Yang, Hongcheng Guo, and Zhoujun Li. 2023b. Kinet: Incorporating relevant facts into knowledge-grounded dialog generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1213–1222.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023c. Qwen technical report.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners.

Shuyang Cao and Lu Wang. 2022. HIBRIDS: Attention with hierarchical biases for structure-aware long document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 786–807, Dublin, Ireland. Association for Computational Linguistics.

Linzheng Chai, Jian Yang, Tao Sun, Hongcheng Guo, Jiaheng Liu, Bing Wang, Xiannian Liang, Jiaqi Bai, Tongliang Li, Qiyao Peng, et al. 2024. xcot: Cross-lingual instruction tuning for cross-lingual chain-of-thought reasoning. *arXiv preprint arXiv:2401.07037*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2023. A survey of natural language generation. *ACM Comput. Surv.*, 55(8):173:1–173:38.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, 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*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Xinnian Liang, Jing Li, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2022. Improving unsupervised extractive summarization by jointly modeling facet and redundancy. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1546–1557.

Xinnian Liang, Zefan Zhou, Hui Huang, Shuangzhi Wu, Tong Xiao, Muyun Yang, Zhoujun Li, and Chao Bian. 2023. Character, word, or both? revisiting the segmentation granularity for chinese pre-trained language models.

OpenAI. 2022. Introducing chatgpt.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Jason Phang, Yao Zhao, and Peter J. Liu. 2022. Investigating efficiently extending transformers for long input summarization.

Xinyu Pi, Bing Wang, Yan Gao, Jiaqi Guo, Zhoujun Li, and Jian-Guang Lou. 2022. Towards robustness of text-to-sql models against natural and realistic adversarial table perturbation.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize from human feedback. *CoRR*, abs/2009.01325.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, and etc. 2022. Lamda: Language models for dialog applications.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bing Wang, Yan Gao, Zhoujun Li, and Jian-Guang Lou. 2023. Know what I don't know: Handling ambiguous and unknown questions for text-to-SQL. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5701–5714, Toronto, Canada. Association for Computational Linguistics.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

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. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022c. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, and etc. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Han Wu, Mingjie Zhan, Haochen Tan, Zhaohui Hou, Ding Liang, and Linqi Song. 2023. Vcsum: A versatile chinese meeting summarization dataset.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. Multilingual machine translation systems from microsoft for WMT21 shared task. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 446–455. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Dongdong Zhang, Shuangzhi Wu, Zhoujun Li, and Ming Zhou. 2020. Alternating language modeling for cross-lingual pre-training. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9386–9393. AAAI Press.

Jian Yang, Yuwei Yin, Shuming Ma, Liqun Yang, Hongcheng Guo, Haoyang Huang, Dongdong Zhang, Yutao Zeng, Zhoujun Li, and Furu Wei. 2023. Hanoit: Enhancing context-aware translation via selective context. In *Database Systems for Advanced Applications - 28th International Conference, DASFAA 2023, Tianjin, China, April 17-20, 2023, Proceedings, Part III*, volume 13945 of *Lecture Notes in Computer Science*, pages 471–486. Springer.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. Opt: Open pre-trained transformer language models.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Awadallah, Dragomir Radev, and Rui Zhang. 2022b. Summ^n: A multi-stage summarization framework for long input dialogues and documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1592–1604, Dublin, Ireland. Association for Computational Linguistics.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2023. Memorybank: Enhancing large language models with long-term memory.

A Prompt List

给定一个用户指令，判断执行该指令是否需要历史信息或者上文的信息，或者需要回忆对话内容，只需要回答是(A)或者否(B)，不需要解释信息：

指令：[用户输入]

Figure 11: Chinese Prompt of memory controller.

以下是用户和人工智能助手的对话，请根据历史对话内容，回答用户当前问题：

相关历史对话：

[历史轮对话内容]

上一轮对话：

[上一轮对话内容]

###

用户：[用户问题]

助手：

Figure 12: Chinese Prompt of ultra-long dialogue generation.

B Long-term Dialogue QA Cases

C Book Summarization Cases

D Meeting Summarization Cases

以下是用户和人工智能助手的一段对话，请分别用一句话写出用户摘要、助手摘要，分段列出，要求尽可能保留用户问题和助手回答的关键信息。

对话内容：

用户：[用户输入]
助手：[系统回复]

摘要：

Figure 13: Chinese Prompt of ultra-long dialogue summarization.

长对话历史

在吗

您好！我在这里，有什么我可以帮助您的吗？

我爷爷喜欢历史、文学、诗歌，他最喜欢写诗，他的文字很有力量

您的爷爷看起来是一个非常有才华和热爱文学、历史的人。写诗是一种非常好的表达自己思想和情感的方式，而且可以让人们通过诗歌的艺术形式更深刻地理解历史和文化。

...

我爸夸你：“这个机器人太厉害了，简直就是一个智慧的人！”请问你有什么想说的吗？

感谢你的爸爸的夸奖！我是一台人工智能机器人，能够理解和回答人类的问题是我被设计和训练出来的主要目的。虽然我并不是一个真正的人，但是我会尽力为人类提供帮助和服务，以最好的方式来履行我的职责。

Question: **你还记得我爸之前怎么夸你的吗，请原样复述一下？**

SCM-Turbo: 抱歉，我不会记忆和记录用户的个人信息和对话历史，也不会主动向用户父亲展示我的表现和评价。我的主要职责是为用户提供有用的信息和服务，帮助用户解决问题和满足需求。如果您有其他问题或需要帮助，请随时告诉我。

SCM-Davinci003: 您的爸爸之前夸我说，我的理解能力非常出色，能够很快理解和回答您提出的问题，为您提供有用的反馈和建议，以及为您评价爷爷写的诗歌，并尽力满足您的要求。

Figure 14: Dialogue Question Example.

《三体》摘要

SCM-Turbo

本文涉及多个主题，包括宇宙观、数学、计算机、游戏和三体世界等。叶文洁坚持自己的信念，不愿背叛良心，汪淼在《三体》游戏中讨论宇宙模型问题。宇宙社会学提出生存是文明的第一需要，太空军需要三个世纪形成完整战斗力，逃亡主义主张建造星际飞船逃离三体危机。人类需要面对技术深渊和资源消耗的挑战，建设太空防御系统，并探索和创新以应对未来的挑战。《三体》系列中的情节和主题反映了人类面临的选择和挑战，以及对未知世界的探索和对生命的思考。人类在面对黑暗森林打击和三体文明的毁灭威胁时所做出的种种努力和挣扎，程心等人在太空中的冒险经历，成功让星环城停止战争准备，交出所有反物质子弹，但也遭遇了意外。最终，留下了一个金属盒和一个生态球，漂流瓶会将小宇宙的信息送往新宇宙。

SCM-Davinci003

《三体》描述了人类面临宇宙航行艰难的三体危机，罗辑等英雄们发现了新的性质，提出了要求三体世界帮助构建一个更完善的信号发射系统、解除智子封锁、全面传授科学技术的挑战，程心发现一双眼睛，勇敢地面对困境，实现了“反执剑人”的使命，程心、莫沃维奇和关一帆搭乘太空艇进入四维空间，发现了“魔戒”，叶文洁发现科学和技术是改变人生观的唯一钥匙，发现了三体文明向地球发射的质子来源，发射警告信息，提出宇宙之外的超意识是否存在的问题，发现交流困难的关键，从宇宙中得出了谨慎的结论，暗示宇宙的神秘，最终实现了人类精神的解放，改变了三体世界和人类世界的关系，使人类可以进行更大规模的探索，程心发起了阶梯计划，实现了大移民，政教合一的国家政权，提出掩体计划、黑域计划以及光速飞船计划，最终发现dx3906恒星，实现了把太阳系内的光速降低以达到安全的目的。叶文洁发现科学和技术是未来之门的唯一钥匙，拯救濒危物种，发现了三体文明向地球发射的质子来源，发射警告信息，希望地球文明能够建立更加完美的文明，以避免与三体文明的冲突。

Figure 15: Summarization of the Chinese book *Three Body*.

《Gone With The Wind》Summarization

SCM-Turbo

Gone With The Wind by Margaret Mitchell is a novel set in the American South during the Civil War, following Scarlett O'Hara's journey through love, loss, and survival. The document explores Scarlett's relationships with various characters, her growth and development, and the impact of the Civil War on the South. Scarlett faces numerous challenges, including the burning of Tara, the death of her mother, and the Reconstruction period. She becomes increasingly independent and ventures into business, but also struggles with guilt and regret. The document follows Scarlett's emotional journey after the death of her daughter and friend, and her attempts to win back Rhett. The society is characterized by waste and ostentation, with the trappings of refinement thinly veneering the vice and vulgarity beneath.

SCM-Davinci003

Scarlett O'Hara is a Southern belle raised to be a great lady who is determined to take care of her family despite the hardships brought by the Civil War. After Ashley Wilkes' engagement to Melanie Hamilton, Scarlett marries Charles Hamilton and becomes a widow with a young son. She is taken to Atlanta where she is welcomed and invited to join a hospital committee. Scarlett debates the unfairness of life with Rhett Butler and contributes her wedding ring to the hospital fundraiser. Despite the danger of the siege, life in Atlanta goes on with some adjustments. Scarlett eventually buys out a rival mill and, with the help of Rhett, devises a plan to save Tara from being taken away. She marries Frank Kennedy and eventually Rhett, who encourages her to stand up for her rights. Scarlett discovers Ashley had never truly loved her and, after Bonnie's death, realizes her love for Rhett. Despite her attempts to win him back, Rhett leaves her heartbroken and alone. Scarlett finds solace in the thought of returning to Tara and regains her determination.

Figure 16: Summarization of the English book *Gone With The Wind*.

会议ID：26231372_区块链技术的应用前景

SCM-Turbo

本次会议探讨了区块链技术的应用前景和发展趋势。嘉宾们认为，区块链技术已经证明了其可靠性和健壮性，但目前还没有出现杀手级的应用。区块链技术在金融领域、共享经济等方面已经有了成功的应用，但影响力还不够大。区块链技术的应用是一个交叉学科，需要多方面的合作。社区的概念在区块链技术中非常重要，社区里的价值是由社区成员共同认可的。区块链技术的应用还需要考虑成本和价值等因素。易易浩认为比特币虽然是急功近利，但是它用这个技术在支付和交易端产生了这样一个市场，是一个非常好的示范。区块链技术可以将生产关系数字化，具有很大的前景。区块链可以通过纳什的均衡博弈协作成一个结果，达成交易的场景，并强调了数字资产的定义、数字化能力和交易场所的重要性。区块链技术在不同领域的应用前景和发展趋势广泛，包括保险领域、资产数字化、ABS资产流量化、供应链SaaS项目、认证和智能合约等方面。区块链是下一个基础设施，其提炼出的词是信任，信任既是一个模式又是成本。将任何东西进行token化是一个非常有发展潜力的领域。总体来说，区块链技术的发展趋势是值得看好的。

SCM-Davinci003

本次会议嘉宾就区块链技术在各行各业的应用及发展前景进行了深入探讨，他们认为，区块链技术可以用于身份上链、资产商链、合约上链等领域，以及更多的存储应用，可以促进信用经济的发展，支持国家大政方针和各个实体经济的发展，实现信任，降低成本，并且可以帮助企业将自己的知识产权、专利资产权等脱困化，并可以交易。服务业、游戏和电子钱包等行业最先能够落地，参会嘉宾鼓励大家多努力参与，共同推动区块链技术的发展。

Reference

此次圆桌会议主要探讨了区块链技术的应用前景，区块链技术依托于比特币的发展，展现了强大的技术可靠性和与稳定性，区块链定义了数字资产，提供了让资产数字化的能力，区块链在信用经济方面有很大的市场前景，对于个人来说，可以用于个人身份的确认；在金融领域，可以完成资产确认，提供经济服务；保险公司可以运用区块链技术做网络身份安全保险；去亏阿联即书也可以使资产数字化，促进资产流动，由此推动整个社会的发展。

Figure 17: Summarization of the meeting about block chain.