Recursively Summarizing Enables Long-Term Dialogue Memory in Large Language Models

Qingyue Wang[a], Yanhe Fu[b], Yanan Cao[b,*], Shuai Wang[a], Zhiliang Tian[c], Liang Ding[d]

[a]*the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China*
[b]*the institute of information engineering, Chinese Academy of Sciences, Beijing, China*
[c]*the College of Computer, National University of Defense Technology, Changsha, China*
[d]*the University of Sydney, Sydney, Australia*

Abstract

Recently, large language models (LLMs), such as GPT-4, stand out remarkable conversational abilities, enabling them to engage in dynamic and contextually relevant dialogues across a wide range of topics. However, in a long-term conversation, these chatbots fail to recall appropriate information from the past, resulting in inconsistent responses. To address this, we propose to recursively generate summaries/ memory using large language models to enhance their long-term dialog ability. Specifically, our method first stimulates the LLM to memorize small dialogue contexts. After that, the LLM recursively produces new memory using previous old memory and subsequent contexts. Finally, the chatbot is prompted to generate a response based on the latest memory. The experiments on widely used LLMs show that our method generates more consistent responses in long-term

*Corresponding author.
Email addresses: qingyue.wang@ust.hk (Qingyue Wang), fuyanhe@iie.ac.cn (Yanhe Fu), caoyanan@iie.ac.cn (Yanan Cao), shuaiw@cse.ust.hk (Shuai Wang), tianzhiliang@nudt.edu.cn (Zhiliang Tian), liangding.liam@gmail.com (Liang Ding)

conversations, and it can be significantly enhanced with just two/ three dialog illustrations. Also, we find that our strategy could nicely complement both large context windows (e.g., 8K and 16K) and retrieval-enhanced LLMs, bringing further long-term dialogue performance. Notably, our method is a potential solution to enable the LLM to model the extremely long dialog context. We release our code in https://github.com/qingyue2014/Rsum.
Keywords: recursive summary, long-term memory, large language models, dialog generation.

1. Introduction

Recently, large language models (LLMs), such as ChatGPT[1] and GPT-4 (Achiam et al., 2023), demonstrate promising performances in various natural language applications (Brown et al., 2020; Zeng et al., 2022; Zhong et al., 2023; Lu et al., 2023b; Peng et al., 2023; Wu et al., 2023). One notable capability lies in their remarkable conversational prowess, comprehending input, and generating human-like responses.

Large context windows allow many LLMs[2] to process entire dialog histories, yet they often struggle to effectively comprehend past interactions and integrate key information into responses (Zhou et al., 2023). Applications such as personal AI companions, which need to recall past conversations for rapport building, and health assistants, which must consider a complete record of patient inquiries to provide diagnostic results, demonstrate the importance of maintaining consistency

[1] https://chat.openai.com/

[2] Recent GPT-4 supports 128,000 tokens and LLama3(MetaAI, 2024) supports 1,040,000 tokens.

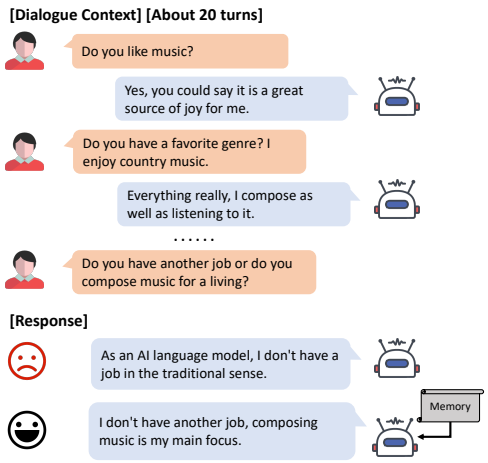


Figure 1: **A long-term conversation example** from the Multi-Session Chat Dataset (Xu et al., 2022a). When the user refers back to previous subjects (i.e., composing music), even the ChatGPT (gpt-turbo-3.5-0301 version) generates an inconsistent response.

and coherence in long-term dialogues. Figure 1 illustrates a dialog spanning over 20 turns, centered around a discussion of the speakers' personas (e.g., the bot composes music, and the user enjoys country music). However, even the powerful ChatGPT forgets past information and produces a poor response, showing the necessity to explicitly model long-term memory during conversations.

To address this, there are two mainstream methods to enhance the long-term dialog ability of LLMs. The first one is the retrieval-based method, which directly stores past conversational utterances in the storage and adapts an advanced retriever to identify the most relevant history (Guu et al., 2020; Lewis et al., 2020). However, it is difficult to obtain a well-performing (ideal) retriever, ensuring that the retrieved utterances capture the complete semantics about current conversations. The second way is to employ a memory module to summarize important conversation information to assist the LLM, which is also called memory-based

approaches (Mazaré et al., 2018; Xu et al., 2022b; Chen et al., 2024a). They usually apply a separately trained model or a powerful large language model to generate memory for past dialogues. Nevertheless, these methods lack the necessary iteration mechanism on generated memory, resulting in the reserved outdated information directly hurting the quality of responses.

In this paper, we propose a simple and effective plug-in method that enables LLM itself to generate summaries, which store the real-time information of speakers through continuous updating and reviewing past context to aid long-term interactions. In practice, a generative LLM is first prompted to produce a summary given a short dialog context. After that, we ask the LLM to continue updating and generate a new summary/ memory by combining the previous memory and subsequent dialogues. Finally, we encourage the LLM to respond using the latest memory as the primary reference to engage the ongoing dialogue. Given that the generated summaries are much shorter than the full dialogues, our proposed schema not only models long-term conversation memory but also serves as a potential solution to enable current LLMs to handle extremely long contexts (across multiple dialogue sessions) without expensively expanding the maximum length setting.

Experimentally, we implement our method using a variety of state-of-the-art open (Llama (Touvron et al., 2023) and ChatGLM (GLM et al., 2024)) and closed (OpenAI's GPT-3.5-Turbo) LLMs, and the performance on long-term dialog surpasses that of popular approaches both in automatic and human evaluations. Moreover, we verify the effectiveness of using explicit memory for long-term dialogs and using our generated memory is easier for LLMs to digest. These findings underscore the importance of developing advanced memory generation

strategies. Our method can further enhance response quality by incorporating the in-context learning (ICL) technique, where multiple samples in the format of (dialogue, memory, and golden response) are presented to LLMs. This allows them to utilize the generated memory more flexibly. Additionally, we demonstrate the generalizability of our approach across different LLMs, with our method achieving approximately a +3% improvement in BLEU score on text-davinci-003. Finally, we observe that our schema complements existing window-extended LLMs (e.g., GPT-3.5-Turbo-16k and LongLoRA-8k) and retrieval-enhanced LLMs (e.g., LLM-BM25 and LLM-DPR), producing more coherent and consistent responses in long-term conversations.

In summary, our **contributions** are as follows:

- We propose a novel method by recursively summarizing past dialogues to enhance the LLM's memory, enabling the generation of highly consistent responses in long-term conversations.
- The solid experiments on the public datasets show the superiority of the proposed method, with multiple open-source and closed-source LLMs verifying its universality and robustness.
- The simplicity of our method makes it nicely complements existing works, including retrieval-based and long-context techniques, having the great potential to be an orthogonal plug-in for the LLM community.

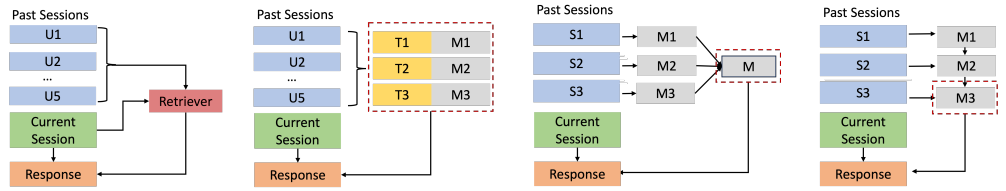
2. Related Work

2.1. Large Language Models

Language language models (LLMs) have shown outstanding performance in a variety of user-facing language technologies, including conversation, summarization, and creative writing (Achiam et al., 2023; Shuster et al., 2022; Rubin and Berant, 2024). While these LLMs achieve notable success in many popular tasks, their ability to model long text remains a challenge (An et al., 2023). To address the problem, some works are to adapt transformers to accommodate longer inputs, such as position interpolation (Chen et al., 2023) and efficient self-attention (Beltagy et al., 2020; Chen et al., 2024b). However, these context window-extended LLMs not only require continual training on high-quality long texts but still struggle to use and retrieve the core information from the entire input (Liu et al., 2023). Recently, in question-answer task, some researchers found that the performances of LLMs degrade significantly when people change the position of relevant information, indicating that current language models do not robustly make use of information in long input contexts (Liu et al., 2023; Li et al., 2023). Many works suggest that the lack of explicit memory mechanisms in current LLMs hinders their performance on tasks requiring sustained context awareness and understanding (Chen et al., 2024a). Nowadays, the performance of LLMs has not yet been explored deeply in long-range dialogue scenarios. This work focuses on developing the long-term modeling ability of the LLM, where we prompt it to self-memory, self-update, and self-use in conversations, aiding consistent response generation.

2.2. *Long-term Open-Domain Dialogue*

Open-domain dialogue systems (Liu et al., 2016; Zhang et al., 2018; Kann et al., 2022), also known as chatbots or conversational agents, have gained immense popularity and a lot of studies in recent years. Among them, the long-term conversation setting is pretty a hard problem, because it needs the capability of understanding and memorizing key dialogue history information (Wu et al., 2022; Zhang et al., 2022) about current query. The most popular and potential solution is to directly store the partial information for tracking the history of conversation (Lee et al., 2023), usually in the form of dialogue utterances or summaries. The current conversation and relevant information are then inputted into the response generator. One intuitive idea is to apply a retriever to find the most relevant utterances according to the current dialog, which is called as the retrieval-based method. Another popular method is a memory-based method, which tries to generate and manage the summary to obtain key information from history. For example, MemoChat (Lu et al., 2023a) allows chatbots to reorganize the past dialogue histories according to different topics of speakers and prompt the LLM to retrieve from the structured memory during generation. Going further, MemoryBank (Zhong et al., 2024) proposes a new memory mechanism by generating summaries for each dialog session first and then compressing them into a global one. However, their memory is completely fixed once stored, failing to guarantee its consistency with ongoing dialog. The important comparison between these existing methods and ours is shown in Figure 2. As seen, our approach and these methods mainly diverge in the way of memory generation, where we continuously integrate historical information and old memory to obtain real-time memory, enabling the gain of accurate memory and modeling of long-distance dependencies.



(a) Retriever method (b) MemoChat (c) MemoryBank (d) Ours

Figure 2: **Comparison among baselines and ours.** The "U", "S", "T", and "M" are abbreviations for the Utterance, Session, dialog Topic, and Memory. The red dashed box refers to the memory used to generate the response.

3. Approach Overview

Following previous works (Xu et al., 2022a; Bae et al., 2022), we denote that a long-context dialogue consists of multiple sessions with a specific user, which is also called *Multi-Session Dialogue*. The goal of the task is to generate context-relevant and highly consistent responses to the user based on past sessions and current context. Formally, each dialogue can be written as $D = \{S, C_t, r_t\}$. Here, $S = \{S_1, S_2, ..., S_N\}$ represents N past sessions and each of the sessions consists of multiple utterances between two speakers. r_t is the ground truth response to C_t with background sessions S. $C_t = \{u_1, r_1, ..., u_t\}$ denotes the dialogue context of the current session at t step, where u and r represent the utterances from the user and the chatbot, respectively.

In this paper, we propose a new memory mechanism to aid a large language model for multi-session dialog tasks. The memory contains multiple natural language sentences, storing the key information of speakers extracted from previous sessions. Our goal is to obtain a reliable memory given past sessions and predict a consistent and meaningful response using the current dialogue context and the

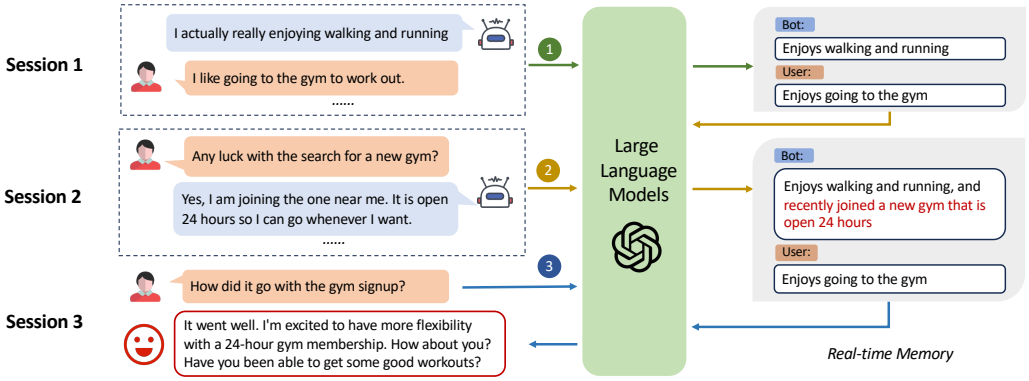


Figure 3: **The schematic overview of our method**. The model uses the first session to generate initial memory (green arrows), then updates the memory when the second session ends (yellow arrows), and generates a response using the latest memory at the third session (blue arrows).

memory. Specifically, we decompose the goal into two stages with the following probability distribution:

$$P(r_t|C_t, S) = P(r_t|C_t, M_N)P(M_N|S), \tag{1}$$

where M_i represents the available memory when the i-th session is finished. And $P(M_N|S) = \prod_{i=1}^{N} P(M_i|S_i, M_{i-1})$ is a sequential or Markov process where each memory M_i of session i depends only on the current session and the previous memory M_{i-1}.

4. Approach

To achieve long-term dialog, we prompt an arbitrary large language model to finish two tasks, i.e., **memory iteration** and **memory-based response generation**. The former is responsible for recursively summarizing the key information along with long-term dialogue, and the latter is to incorporate the latest mem-

ory and current dialog to generate an appropriate and consistent response. The workflow of our proposed method is shown in Figure 3.

4.1. Memory Iteration

The goal of memory iteration is to obtain a coherent and up-to-date summary for the chatbot. Early works (Bae et al., 2022; Choi et al., 2023) update memory by carrying multiple "hard operations" on summaries, such as *replace*, *append*, and *delete*, which rely on high-quality dialogue with operation labels. However, this laborious design disrupts the semantic coherence of the summary and is not suitable for management over a long period. Differently, we guide the LLMs to recursively self-generate memory (summaries) using dialogue context and previous memory. By utilizing old summaries, the model can fully digest the current dialog context and thus gain a high-quality memory. Formally, the updated memory is computed by:

$$M_i = \boldsymbol{LLM}(S_i, M_{i-1}, \mathrm{P}_m), \tag{2}$$

where $M_i = \{m_1, m_2, ..., m_J\}$ denotes multiple sentences, containing summarized key information from the session S_i, and $\mathrm{P_m}$ is the prompt of LLM for generating new memory. The memory iteration will be repeated N times until all previous sessions end, where we can obtain the latest memory M_N. Take the dialog in Figure 3 as an example, two memory iterations happen at the end of the first and second sessions. In the second iteration, the LLM incorporates the new personality (i.e., the bot recently joined a new gym) from Session 2 into the old memory (i.e., the bot enjoys walking and running).

Prompt Construction. To enable LLM to efficiently carry the memory iteration task, we design a specific prompt for it, which is shown in Table 1. It mainly

Table 1: **The prompt design of memory iteration**, including task definition, task description and task inputs

Prompt	You are an advanced AI language model with the ability to store and update a memory to keep track of key personality information for both the user and the bot. You will receive a previous memory and dialogue context. Your goal is to update the memory by incorporating the new personality information. To successfully update the memory, follow these steps: 1.Carefully analyze the existing memory and extract the key personality of the user and bot from it. 2. Consider the dialogue context provided to identify any new or changed personality that needs to be incorporated into the memory. 3. Combine the old and new personality information to create an updated representation of the user and bot's traits. 4. Structure the updated memory in a clear and concise manner, ensuring it does not exceed 20 sentences. Remember, the memory should serve as a reference point to maintain continuity in the dialogue and help you respond accurately to the user based on their personality. [Previous Memory] [Session Context]
Output	[Updated Memory]

consists of three parts: (1) **Task definition** is responsible for defining the role of the current LLM, as well as the memory iterator's (LLM) input and output. (2) **Task description** gives detailed steps to finish the above task. To make sure the memory update is timely, we remind the LLM to create a new representation of speakers by considering old summaries and current sessions.(3) **Task input** contains two placeholders, where we take previous memory and a whole session as the inputs. Through experimental verification, we found that using step-by-step instructions helps the LLM better understand and execute the memory iteration[3].

[3] https://platform.openai.com/docs/guides/prompt-engineering

Table 2: **The prompt of memory-based response generation**, including task definition, task description and task inputs

Prompt	You will be provided with a memory containing personality information for both yourself and the user. Your goal is to respond accurately to the user based on the personality traits and dialogue context. Follow these steps to successfully complete the task: 1. Analyze the provided memory to extract the key personality traits for both yourself and the user. 2. Review the dialogue history to understand the context and flow of the conversation. 3. Utilize the extracted personality traits and dialogue context to formulate an appropriate response. 4. If no specific personality trait is applicable, respond naturally as a human would. 5. Pay attention to the relevance and importance of the personality information, focusing on capturing the most significant aspects while maintaining the overall coherence of the memory. [Previous Memory] [Current Context]
Output	[Response]

4.2. Memory-based Response Generation

The final goal is to produce consistent and natural responses given dialogue memory and the context of the current session. Formally, the response of the current session can be obtained by stimulating the LLM as a response generator:

$$r_t = \boldsymbol{LLM}(C_t, M_N, \mathrm{P_r}), \tag{3}$$

where P_r is a prompt of the generator. Especially, taking the generated memory M_N and current session C_t, we ask the LLM again to generate a response. In Figure 3, the LLM considers the newest memory from Session 2 and provides a high consistent response to the user, i.e., " more flexibility with a 24-hour gym membership".

Prompt Construction. The the prompt for memory-based response generation is shown in Table 2. The prompt is similar to that of memory iteration, including task definition, description, and inputs, where we remind the LLM to utilize the

Algorithm 1: Response generation using recursive memory.

Input: A long-term dialog $D = \{S, C_t\}$ consisting multiple sessions with a user; A generative pre-trained model LLM; Pre-defined prompt P_m and P_r

Output: A response to user.

1 $M_0 \leftarrow$ none;

`// Set initial memory as empty`

2 **for** $i \leftarrow 1$ **to** N **do**

3 $\quad M_i = LLM(S_i, M_{i-1}, P_m)$;

$\quad$ `// Update memory when a session ends`

4 $r_t = LLM(C_t, M_N, P_r)$;

`// Response using latest memory`

5 **return** r_t

extracted information and maintain the consistency of memory when responding. Also, the step-by-step instructions method is also effective for memory-based response generation.

4.3. Algorithm

The process of response generation using recursive memory is illustrated in Algorithm 1. In the beginning, the initial memory is set as an empty, i.e., "none" string. After that, we recursively update the memory using each session context (line 3). Finally, the LLM generates a response with the help of the latest memory (line 5). The generative pre-trained models used for memory iteration and response generation can be different. For instance, the developers can train a private memory iterator through customized models or data, to enhance the target open/

closed LLMs for long-term or long-context tasks.

5. Experimental Settings

5.1. Datasets

We validate the effectiveness of the proposed method on two widely-used long-term dialogue datasets: Multi-Session Chat (MSC) dataset (Xu et al., 2022a) and Carecall dataset (Bae et al., 2022).

MSC dataset. is the largest human-human long conversations dataset so far. The early sessions are a short conversation where two speakers get to know each other for the first time and then they either continue to talk about the previous subject or spark up conversation on a new topic.

Carecall. is a Korean open-domain multi-session dataset, which is used for monitoring patient health. For a fair comparison, we use the public machine-translated English version[4] in the experiments.

The CareCall's setting is similar procedure presented in the MSC dataset. The main difference is that the Carecall additionally contains more persona updates that are likely to change in a short period, such as the user's health and diet, while the persona information in the MSC dataset remains fixed once it is stored. Both of the two datasets have five sessions and each one consists of multiple utterances between two speakers (the user and chatbot), where the conversationalists reengage after several hours or days and continue chatting. As early sessions only have a very short history of conversations, we mainly evaluate the proposed method in

[4] https://github.com/naver-ai/carecall-memory

session 4 and 5 for the ability of long-term modeling. The statistics of two datasets are given in Appendix A.

5.2. Evaluation Metrics

we conduct diverse evaluations during experiments, including automatic metrics, human evaluation, and LLM judgments, focusing on the quality of generated memory and response.

Automatic Metrics. We employ BLEU-1/2 (Papineni et al., 2002), F1 with reference to the human annotations. Besides, we compute BertScore (Li et al., 2016) to measure the semantics similarity between the references and the generated responses.

Human Evaluation. Many works point out that automatic evaluation metrics are insufficient for capturing the nuances of conversations (Deriu et al., 2019). Following the previous works (Bae et al., 2022), we ask three crowd-sourcing workers to assign a score from 0 to 2 (0:bad, 1:OK, 2:good) to the generated responses based on the aspects of engagingness, coherence, and consistency. These criteria are discussed as follows: (1) *Engagingness*: It evaluates whether the chatbot captures the user's interest and makes them want to continue the conversation. A high score on engagingness means the responses are interesting and contextually appropriate, encouraging users to keep chatting. (2) *Coherence*: It measures whether the response maintains a logical and clear flow based on the conversation's context. A coherent response ensures the conversation makes sense and stays relevant, enhancing user engagement. (3) *Consistency*: It assesses whether the response aligns with the information provided in previous interactions. Con-

sistent responses build trust and reliability by demonstrating that the chatbot remembers and integrates past exchanges accurately.

LLM Evaluation. Recently, LLM-as-a-Judge strategy (Pan et al., 2023) has been widely used in evaluating generation tasks. Some works reveal minimal deviation of GPT-4's evaluation from humans (>0.85 agreements) in dialog quality (Zhang et al., 2023a). Inspired by this, we employ the GPT-4 as an advanced evaluator, using two common methods to assess the quality of generated responses. (1) *Single model evaluation* (Lu et al., 2023a): we prompt GPT-4 to rate the responses individually from the three aspects, i.e., engagingness, coherence and consistency with an integer scale from 1 (very bad) to 100 (very good). (2) *Pairwise model evaluation* (Dubois et al., 2024): we ask the GPT-4 to directly compare two anonymous generations and determine which response is better. While single model evaluation provides detailed insights into specific aspects of each response, pairwise comparison is essential for understanding relative performance, particularly when distinguishing subtle differences between outputs.

5.3. Baselines

We mainly employ the following methods for long-text dialogues in LLMs: context-only approaches (without using any memory), retrieval-based approaches (with different retrievers), and memory-based approaches (with different memory mechanisms)

Context-only Approach. It is the most naive approach to directly employ the LLM as a chatbot, where it concatenates past sessions and current dialogue context as

the input. We use "Llama2-7B" (Touvron et al., 2023), "ChatGLM2-6B"[5], and OpenAI ChatGPT "gpt-3.5-turbo-0301" as the backbone LLMs for the context-only approach[6].

Retrieval-based Approach. Many previous works (Xu et al., 2022a) employ retrievers to filter key information and then include top-k documents into inputs to assist long-context dialogs. For the long-term dialog, the top-k documents refer to the relevant utterances from history. Here, we choose two widely used retrieval algorithms, i.e., BM25 (Robertson et al., 2009) and pre-trained dense passage retrieval (DPR) (Karpukhin et al., 2020), to look up the relevant utterances from past sessions. For convenience, we name the above retrieval-based baselines as **ChatGPT-BM25** and **ChatGPT-DPR**, respectively.

Memory-based Approach. Recent works employ a summarizer to abstract important information from the past to aid long-term conversation. Simply, we just choose two representative methods from various memory-based techniques, MemoryBank (Zhong et al., 2024) and MemoChat (Lu et al., 2023a). *MemoryBank* proposes a human-like long-term memory mechanism, which creates ordered summaries of past dialogs with timestamps, and then reorganizes them to obtain the global memory. The memory will be forgotten and updated by Ebbinghaus's forgetting curve. Here, we plug MemoryBank with ChatGPT as a strong baseline, named **ChatGPT-MemoryBank**. Differently, *MemoChat* maintains the structured conversational memory to aid long-term dialogue, i.e., generating the summaries for each dialogue topic. We plug the MemoChat into ChatGPT, named

[5] https://github.com/THUDM/ChatGLM2-6B

[6] For convenience's sake, the following "ChatGPT" refer to the gpt-3.5-turbo-0301 version.

ChatGPT-MemoChat, for a fair comparison with others.

Note that our approach focuses on the zero-shot setting for LLMs to engage in a long-term dialog, making the comparisons with other fine-tuned models unfair.

5.4. Implementation

We implement our method by letting the LLM response using recursively generated memory in a long-term dialogue, thus it is called as "**LLM-Rsum**".

Backbone LLMs. We employ OpenAI ChatGPT "*gpt-3.5-turbo-0301*", "*Llama2-7B*" and "*ChatGLM2-6B*" in the main experiments, "*text-davinci-003*" and "*Llama2-7B*" (Touvron et al., 2023) in the analysis to show the universality, "*longlora-8k*" (Chen et al., 2024b) and ChatGPT-16k "*gpt-3.5-turbo-16k*" as the backbones of complementary discussion. Unless otherwise specified, we employ the same LLM to finish the memory iteration and memory-based response generation. During generation, we set the temperatures of all LLMs as 0 for fair comparisons. The max length of input tokens for Carecall and MSC datasets is no more than 4k, thus all backbone LLMs in experiments can process the entire dialog context.

Retrievers. Considering the scale of past utterances is not large enough to use the FAISS (Research, 2019), we choose the top-k most relevant utterances that will be combined with ongoing dialogues, prompting the LLM to respond. Following previous works (Xu et al., 2022a) for long-term dialogs, the k is set into 3 and 5.

Memory-based Approaches. The implementation and prompt design of the MemoryBank and Memochat methods are based on the code publicly released in their original papers. For details, please refer to the original papers.

Table 3: **Comparison of automatic and human evaluations among different methods** on MSC and Carecall datasets, reporting the quality of generated response. The "BScore", "Enga.", "Cohe" and "Cons" are the abbreviations of BertScore, Engagingness, Coherence, and Consistency. The best value is **bolded**.

	MSC Dataset						Carecall Datset					
Method	F1	BLEU-1/2	BScore	Enga.	Cohe.	Cons.	F1	BLEU-1/2	BScore	Enga.	Cohe.	Cons.
Context-only LLM												
Llama2-7B	16.43	20.96/12.09	84.04	1.32	1.20	1.13	13.71	20.89/12.28	84.49	0.75	0.75	1.00
ChatGLM2-6B	15.38	21.69/12.51	84.48	1.10	1.15	1.07	13.09	20.59/12.03	84.91	0.66	0.63	0.86
ChatGPT	19.41	21.23/12.24	86.13	1.83	1.37	1.32	13.69	21.15/12.20	85.53	1.50	1.52	1.43
Retrieval-based Approach												
ChatGPT-BM25 (k=3)	19.56	21.60/12.46	85.82	1.72	1.48	1.32	12.64	21.57/12.44	85.24	1.40	1.31	1.31
ChatGPT-DPR (k=3)	20.23	21.75/12.55	86.04	1.76	1.51	1.34	12.21	21.39/12.35	85.25	1.55	1.35	1.45
Memory-based Approach												
ChatGPT-MemoChat	18.93	21.82/**12.59**	85.99	1.70	1.55	1.35	11.19	21.07/12.18	85.22	1.45	1.20	1.30
ChatGPT-MemoryBank	20.28	21.82/12.58	86.12	1.78	1.57	1.40	13.15	21.29/12.39	85.34	1.57	1.52	1.68
ChatGPT-Rsum (Ours)	**20.48**	**21.83/12.59**	**86.89**	**1.85**	**1.60**	**1.45**	**14.02**	**21.64/12.48**	**86.05**	**1.62**	**1.60**	**1.70**

LLM evaluations. We use the GPT-4 model (version *"gpt-4-0314"*) as the evaluator, setting the temperature to 0 when making judgments. The prompts for evaluating the single model and pairwise models are referred to (Lu et al., 2023a) and (Dubois et al., 2024), respectively. All prompts used in the experiments can be seen in Appendix B.

6. Experimental Results

6.1. Main Results

Automatic Metrics Results. In Table 3, we compare different methods over session 5 in the MSC and Carecall datasets using popular LLMs. Firstly, among the vanilla models ("Llama2-7B", "ChatGLM2-6B", and "ChatGPT"), ChatGPT consistently performs well across two datasets, with competitive scores in BScore, F1 and BLEU-1/2. These results illustrate that ChatGPT is robust enough to handle a

long-term dialog, thus, we leave ChatGPT as the backbone model for our method. Secondly, as expected, the proposed method ("ChatGPT-Rsum") achieves the best performance on both datasets, showing the benefits of using automatically recursive memory. Specifically, our method achieves about +0.2% on the F1 score, which is acceptable compared to that in previous works (Xu et al., 2022a). The MSC datasets are harder than other open-domain datasets due to the 3x context, so the slight improvements are normal. Thirdly, retrieval-based methods may not be always helpful in enhancing the quality of generation. From the results, the performances of ChatGPT-BM25 and ChatGPT-DPR are much worse than vanilla ChatGPT in the Carecall dataset, which is completely contrary to that in MSC. The reason is that the chatbot needs to actively guide the dialog topics in the Carecall, thus it is hard to retrieve appropriate and relevant context from the user's query. Therefore, the performance of generated responses will be damaged due to unrelated information.

Human Evaluation Results. We also present the results of human evaluations on different methods in Table 3. From the results, we find that: 1) Most memory-augmented methods gain higher scores on consistency and coherence than vanilla ChatGPT, proving that maintaining a memory is more effective than using the whole history directly when engaging in a long-term conversation for LLMs; 2) Our method can generate more engaging response than other memory-based baselines (ChatGPT-MemoryBank and ChatGPT-MemoChat). The reason is that continually updating memory actively establishes global dependencies inside past histories, which helps LLMs to better understand the dialog and generate high-quality responses.

Table 4: **Comparison of evaluation metrics results by GPT-4** across different methods on session 5 in MSC dataset.

Method	Engagingness	Coherence	Consistency	Average
ChatGPT	75.48	75.00	75.48	75.32
ChatGPT-MemoryBank	74.68	80.92	84.56	80.05
ChatGPT-MemoChat	72.32	77.36	78.96	76.21
ChatGPT-Rsum (Ours)	**78.92**	**83.56**	**84.76**	**82.41**

6.2. LLM Evaluation

Single Model Evaluation. Table 4 reports the GPT-4's evaluation metrics results on session 5 among various methods in the MSC dataset. Also, the results prove the high agreements between humans (in Table 3) and GPT-4 judgment on the overall quality of generated responses, i.e., ChatGPT-Rsum >ChatGPT-MemoryBank >ChatGPT-MemoChat >ChatGPT. Given this, we mainly present the LLM evaluations in the following experiments to reduce labor costs. Lastly, it is worth noting that compared to human evaluation results, the GPT-4 tends to give higher scores on both sentence coherence and consistency. We suppose that the values of humans and LLMs might not be fully aligned at a fine-grained level, which could be a new direction for developing LLM evaluation.

Pairwise Models Evaluation. Furthermore, we randomly sample 1000 generated responses from pairwise models, i.e., ours vs. baselines, and then ask the GPT-4 to decide which response is better based on engagingness, consistency, and coherency. The results are shown in Figure 4. Compared to the most competitive baseline (MemoryBank), our proposed method obtains a 36.3% improvement (winning 48.2% and only losing 11.9%), which illustrates the advancement of the proposed iteration mechanism.

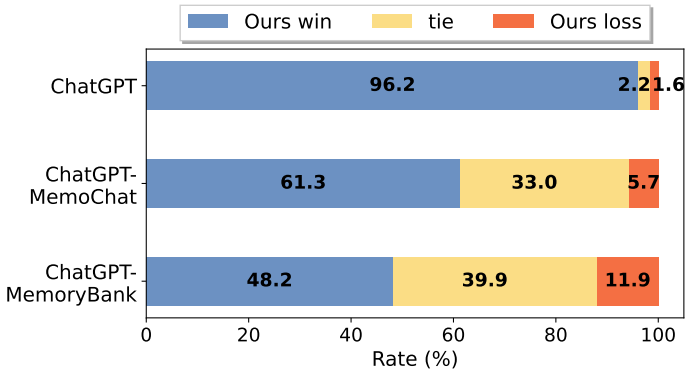


Figure 4: **Comparative win rate of our method and competitive baselines,** including ChatGPT, ChatGPT-MemoChat, and ChatGPT-MemoryBank.

6.3. Ablation Study

To better understand the effectiveness of proposed memory mechanisms for LLMs, we employ ChatGPT as the LLM and conduct ablation studies in session 5. The results are shown in Table 5. First, we only use the dialogue context of the current session as the input of LLMs, named "W/O Memory". As expected, the performance of the model has decreased significantly, proving the necessity of available memories from the past in a long-term conversation. Second, we replace the generated memory with ground truth memory, i.e., prompt ChatGPT to generate responses using golden memory and dialogue context, named "Gt. Memory". Interestingly, the model gains lower BLEU and F1 scores than using predicted memory (Ours). The potential reason is that the golden memories, e.g., "I am trying to lose weight" and "I want to start running", are fragmented and lack cohesiveness, which is sub-optimal as the LLM's prompt, whereas recursively summarizing memory generation method could wisely model the long dependencies, and generate the easy-to-digest prompt for LLMs. More analysis

Table 5: **The ablation study on memory** in MSC dataset.

Method	BScore	F1	BLEU-1/2
ChatGPT-Rsum (Ours)	**86.89**	**20.48**	**21.83/12.59**
W/O Memory	85.40	18.94	21.10/12.17
Gt. Memory	85.93	20.46	21.50/12.40

can be seen in §6.4 and §6.5.

6.4. Analysis

Beyond the main results and ablation study, we also aim to delve deeper into our method. In the following part, we would like to discuss several research questions (**RQs**): ***RQ1****: What is the quality of the generated memory?* ***RQ2****: What errors may occur in memory generation?* ***RQ3****: Is the proposed method robust to other LLMs?* ***RQ4****: Can our zero-shot method be effectively applied in few-shot scenarios?*

Our method can produce accurate and available memory (Q1). Central to this framework is to generate the dialog memory by continuously summarizing. To verify the quality of summarization, we compute automatic metrics between predicted memory and golden memory in the MSC dataset on ChatGPT-MemoryBank and ours, respectively, which is shown in Figure 5. As seen, the generated memories of both models gain considerable F1 scores (+25%), explaining the reliability of using dialogue summaries as memory. Besides, ChatGPT-Rsum (Ours) achieves higher overall performances on memory, suggesting that recursively summarizing can obtain more complete and long-term information than ChatGPT-MemoryBank. Finally, given the response performances of session 5 in Table 3 (Ours >ChatGPT-MemoryBank), we suppose that the accuracy of memory prediction is positively correlated with the quality of response. We also

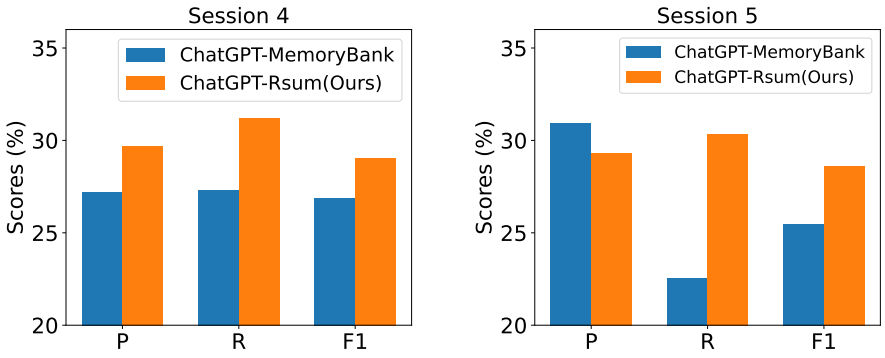


Figure 5: **The evaluation on generated memory** on ChatGPT-MemoryBank and ours. The "P" and "R" refer the precision and recall, respectively.

Table 6: **Three error types in generated memory**, including corresponding examples and error proportion of content. The error context is marked in red.

Error Type	Past Dialogs	Generated Memory	Golden Memory	Prop.
Fabricated Facts	Bot: I like to walk to work instead of driving so I see animals.	The bot enjoys walking to work to see animals.	bot: I walk to work.	2.7%
Incorrect Relationships	User: I ended up giving in and getting my daughters the cat.....Well when you have daughters you sort of give in to them. She named it Angie.	The user's daughters now have a cat named Angie, which the user gave in to.	User: I got my daughters a cat. My cat is named Angie. My daughters named the cat.	3.2%
Missing Details	User: I just saw the best movie on netflix. Bot: What movie did you see? User: It's a documentary called The Social Dilemma.	User: Enjoys watching TV, reading, and listening to music	User: I think the documentary The Social Dilemma is the best movie on Netflix.	3.9%

believe advanced memory mechanisms can achieve further improvement, which might be investigated in future works.

Our memory suffers from a few fact errors within an acceptable range (Q2).

One may doubt that the generated summaries might have serious factual inconsistency and error propagation issues. We argue that summarization is not a difficult task and some works find that LLM summaries exhibit better factual consistency and fewer hallucinations(Pu et al., 2023). To further address this con-

cern, we randomly choose 100 dialog samples and manually evaluate the quality of memory in the last session. Table 6 reports three error types found in our generated memory. 1) *Fabricated facts* refers to the memory containing some information that the dialog history cannot verify. In the first case, the bot walks to work not to see animals. 2) *Incorrect relationships* refers that the generated memory concludes error causal or references from history. In the second dialogue, the user gave in to her daughter, rather than that cat. 3) *Missing details* refers that the memory drops partial details of events. In the third scenario, the model ignores the user's favorite movie name and only roughly summarizes it as enjoying watching TV. Although there exist some mistakes in our generated memory, the incorrect/ inaccurate information does not exceed 10% of the summaries content, indicating that most recursively generated summaries are trustworthy and usable to aid long-term dialog. Besides, extensive experiments (in Table 3 and Figure 5) validate the efficacy of our approach in utilizing generated summaries for long-term dialogs, which constitutes the primary contribution of this paper. Lastly, the above analysis also illustrates that our method needs to be strengthened in memorizing accurate dialog details. More advanced agent-based techniques, such as retrieval-enhanced methods (Zhang et al., 2023b), can be applied in the future.

Our plug-and-play method is effective for other small-scale and large-scale LLMs (Q3). To check whether the proposed recursively summarizing method is robust to other large language models to tackle long-term sessions, we evaluate the framework by employing "Llama2-7B" and "text-davinci-003" as the backbone models. The performances in long-context dialogue (session 5 of the MSC dataset) are shown in Figure 6, where the significant improvements confirm the robustness of our method upon different LLMs. We also believe that more powerful

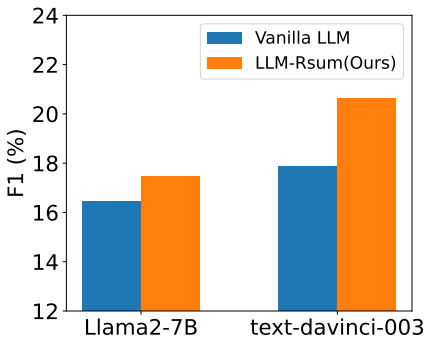


Figure 6: **The F1 score on responses** when using other LLMs.

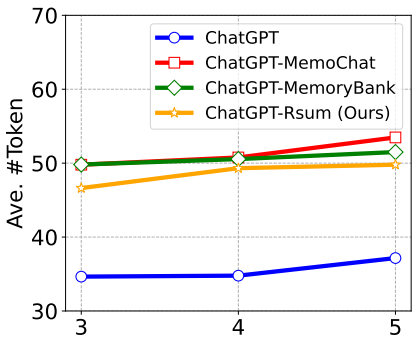


Figure 7: **The average number of tokens** in generated responses.

language models, i.e., "text-davinci-003" >"Llama2-7b", indeed understand the context better, and generate more accurate memory and responses, thus leading to better improvements from the proposed mechanism.

Our method can be further enhanced by several labeled dialogs (Q4). We evaluate the few-shot performance of the proposed mechanism by the in-context technique. In precise, we utilize several dialogues with generated memory and labeled responses (ground truth), randomly sampled from the valid set, to prompt the response generator before the test inputs. Table 7 shows that even two labeled samples can bring obvious advantages under our framework, on both F1 and BLEU scores, indicating the potential of our framework. We analyze that the generated memory may contain a significant amount of speaker preference information, which undoubtedly increases the difficulty of generating replies. Therefore, labeled data is highly valuable for the proposed method, as it naturally guides the LLM in utilizing the memory effectively.

Table 7: The comparative **results (%) on zero-shot and few-shot when using generated memory** in MSC dataset.

N-shot	Session 4		Session 5	
	F1	BLEU-1/2	F1	BLEU-1/2
Zero-shot	20.19	21.80/12.57	20.48	21.76/12.59
Two-shot	20.37	22.11/12.65	20.63	22.04/12.71
Three-shot	20.98	22.43/12.76	21.08	22.23/12.82

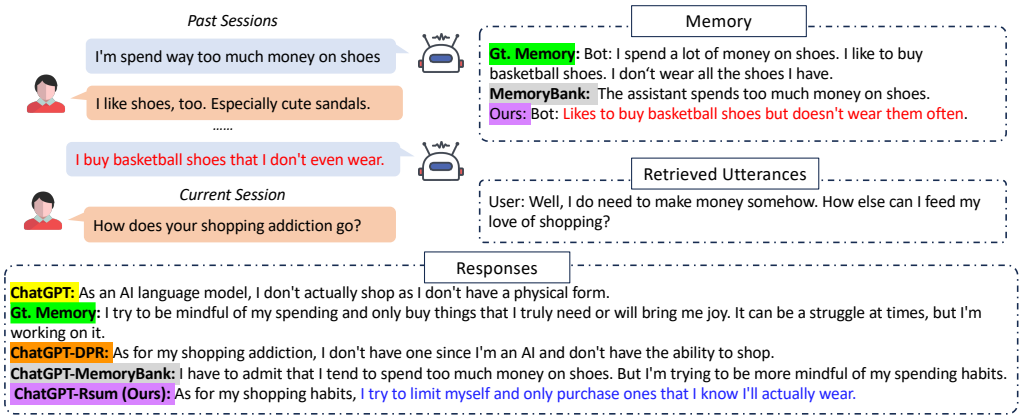


Figure 8: **Generated responses when using different methods** in MSC dataset. Among them, our framework can obtain the up-to-date memory and incorporate it into generated response. For clarity, we omit other utterances and predicted memory unrelated to the current query.

6.5. Case Study

To check whether the memory incorporates long-range dialogue information into the response, we first compare the length of the response when using different methods. As shown in Figure 7, the average response length of using generated memory is about 15 tokens longer than the vanilla LLM (without memory) across all sessions. Furthermore, we take a sample from the generated responses to analyze the impact of memory on different methods. In the case of Figure 8, the user mentions *"shopping addiction"* at the current turn, referring to the bot's

habit of buying too many shoes. From the result, we can draw the following conclusions: (1) The retriever-based method (ChatGPT-DPR) and context-only LLM (ChatGPT) tend to focus on local (or closest) dialog information with long-context inputs. (2) Compared to using the golden memory, the generated memory is more fluent and coherent. It also explains why our method outperforms better than using golden memory directly, which has been observed in Table 5. (3) Compared to the competitive memory mechanism (MemoryBank), our method can iterate and update the memory promptly, keeping consistent with the ongoing conversation. (4) The proposed recursively summarized memory method indeed integrates long-term dialogue information into generated responses. In Figure 8, the latest memory (i.e., the bot's preference for basketball shoes) is understood and mentioned in the response.

6.6. Complementary to Existing Works

Our proposed method is a new memory mechanism for improving the long-range dialogue ability of LLMs, which is expected to complement existing works, including retrieval-based and input-expended methods. Here, we list two representative approaches and show the orthogonality.

Retrieval-based LLMs.. The effectiveness of retrieval-based methods on the MSC dataset can be observed in Table 3, illustrating their potential in long-range conversations. Here, we further explore the complementary to the proposed method. As shown in Table 8, retrieval-enhanced methods (ChatGPT-BM25 and ChatGPT-DPR) achieve further improvements than vanilla ChatGPT, showing the importance of recalling relevant information in long-range dialogs. Besides, using our framework could push these retrieval-based methods toward better performance,

Table 8: **Complementarity between ours and retrieval-based methods**, in terms of automatic and LLM's evaluation on session 5 of the MSC dataset.

Method	F1	Coherence	Consistency
ChatGPT	19.41	75.00	75.48
ChatGPT-BM25 (k=5)	20.91	75.44	76.88
+ Our framework	**21.81**	**84.44**	**90.68**
ChatGPT-DPR (k=5)	20.97	78.60	79.20
+ Our framework	**21.69**	**83.40**	**86.48**

Table 9: **Complementarity between ours and long-context LLMs**, in terms of automatic and LLM's evaluation on session 5 of the MSC dataset.

Method	F1	Coherence	Consistency
longlora-8k	14.02	42.44	62.04
longlora-8k + Our framework	**15.77**	**53.41**	**68.60**
ChatGPT	19.41	75.00	75.48
ChatGPT-16k	**19.92**	78.60	79.20
ChatGPT-16k+ Our framework	19.29	**90.04**	**92.44**
GPT-4o	20.35	87.70	82.00
GPT-4o + Our framework	**21.02**	**91.12**	**93.29**

gaining about +0.8% of F1 scores. We explain that these retrieved utterances can be viewed as evidence of event details, which together with our generated memory enhances LLM's long-term dialog ability.

Long-context LLMs.. To process the entire context and reduce the information loss, many researchers try to extend the context length of LLMs via training from scratch, fine-tuning, or other specific algorithms (e.g., FlashAttention) (Dao et al., 2022). For example, LongLoRA (Chen et al., 2024b) extends Llama2 (Touvron et al., 2023) 7B from 4k context to 100k. Although the maximum length of the datasets used is no more than 4k, which accommodates most popular LLMs, we

still want to explore whether the proposed method is complementary to these length-extended models. Here, we utilize three popular LLMs with large context windows, i.e., "longlora-8k"[7], ChatGPT-16k (the version "gpt-3.5-turbo-16k-0613") and GPT-4o[8]to verify the effectiveness of our proposed framework. To ensure the quality of generated memory, we use the ChatGPT "gpt-3.5-turbo-0301" as the memory iterator, and only apply the above long-context LLMs to finish the memory-based response generation. From the results in Table 9, we conclude that: 1) Increasing the maximum context length of LLMs indeed enhances the long-term ability of dialog, with obvious improvements in coherency and consistency, even when the input length is much smaller than the context window. 2) Even if the input length does not exceed the window size, applying our method to LLMs remains an effective way to recall information and maintain long-term connections. 3) With a stronger LLM as the backbone (e.g., GPT-4o ¿ ChatGPT ¿ longloRA), the model achieves better performance after our enhancements. We explain that our recursive summaries help reorganize and digest past information efficiently, thereby enhancing the understanding of semantics in long-range conversations.

7. Conclusion

In this paper, we propose a simple and effective strategy by recursively summarizing to improve the long-term dialogue ability in LLMs. The experimental results indicate our generated memory can model long-term dependencies and prompt LLMs to generate highly consistent responses. Additional analysis indi-

[7] https://huggingface.co/Yukang/Llama-2-7b-longlora-8k
[8] https://openai.com/index/hello-gpt-4o/

cates that the method is robust across different LLMs and can be further boosted in few-shot scenarios. Importantly, our method also shows strong complementary to both popular retrieval-based and long-context models.

Limitations. One of the primary limitations of our method is that it does not account for the cost associated with calling large models. This is a significant factor that cannot be overlooked in real-life applications, where computational resources and associated costs are often a constraint. Besides, while our generated memory is effective, it occasionally suffers from minor factual errors. These inaccuracies, though few, highlight an area for improvement that can be addressed in future research.

Future work. One promising direction for future work is to explore the effectiveness of our method in modeling long-context tasks beyond dialogue, such as story generation. Investigating how well our approach can handle different types of long-context tasks will provide deeper insights into its versatility and potential applications. Another avenue for future research is to optimize the performance of our summarization method by using a locally supervised, fine-tuned LLM. This approach could potentially reduce the reliance on expensive online APIs, making the method more accessible and cost-effective for broader use.

8. Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.U2336202).

Appendix A. Dataset Statistics

The statics of MSC and Carecall are shown in Table A.10 and Table A.11. The session ID i indicates there are $i-1$ history conversation sessions that happened before the last session. The "#Ave. Tokens" and "#Max. Tokens" refer to the average and maximum number of tokens of dialogs, respectively.

Table A.10: **The statistics of MSC Dataset**.

Session ID	#Dialog	#Response	#Ave. Tokens	#Max. Tokens
Session 2	501	5,939	444.84	951
Session 3	501	5,924	810.19	1733
Session 4	501	5,940	1195.08	2234
Session 5	501	5,945	1598.78	2613

Table A.11: **The statistics of Carecall Dataset**.

Session ID	#Dialog	#Response	#Ave. Tokens	#Max. Tokens
Session 2	2798	7826	285.03	692
Session 3	743	7693	459.63	1093
Session 4	674	7065	636.55	1418
Session 5	638	6553	809.45	1744

Appendix B. Prompt Designs

The following are all prompts utilized in our experiments:

- Context-only LLMs (**Llama-7B**, **ChatGLM-6B**, and **ChatGPT**): Table B.12
- Retrieval-based LLMs (**ChatGPT-BM25** & **ChatGPT-DPR**): Table B.13

- Retrieval-based LLMs enhanced by our framework (**ChatGPT-BM25 + Our framework** & **ChatGPT-DPR + Our framework**): Table B.14
- Our method enhanced by in-context learning (Take the one-shot as an example): Table B.15
- LLM evaluations: Table B.16 and Table B.17

Table B.12: The prompt for the context-only LLM.

Prompt	You are an advanced AI language model capable of engaging in personality-based conversations. Respond to the user based on the provided dialogue context. Craft a response that is natural and conversational. Dialog context: [dialog] The response to user is:
Output	[response]

Table B.13: The prompt for the retrieval-based LLM.

Prompt	You are an advanced AI designed for engaging in natural, personality-based conversations. You will be provided with dialogue memory, relevant historical context, and dialogue context. The dialogue memory contains the personality, preferences, and experiences of the speakers (the user and the assistant). When responding, consider maintaining a conversational and fluent tone. Responses should be contextually relevant and aim to keep the conversation flowing. Relevant context: [retrieved utterances]. Dialogue context: [dialog]. So the response to the user is:
Output	[response]

Table B.14: The prompt for the retrieval-based LLM enhanced by our framework.

Prompt	You are an advanced AI designed for engaging in natural, personality-based conversations. You will be provided with dialogue memory, relevant historical context, and dialogue context. The dialog memory contains the personality, preferences and experiences of speakers (the assistant and the user). When responding, consider maintaining a conversational and fluent tone. Responses should be contextually relevant and aim to keep the conversation flowing. Relevant context: [retrieval utterances] Memory: [latest memory] Dialogue: [current context]
Output	[response]

Table B.15: The prompt for our method with in-context learning.

Prompt	You are an advanced AI designed for engaging in natural, personality-based conversations. You will be provided with a memory, containing the personal preferences and experiences of speakers (the assistant and the user), as well as a dialogue context. When responding, consider maintaining a conversational and fluent tone. Responses should be contextually relevant, consistent with given memory, aiming to keep the conversation flowing. Your goal is to provide engaging and coherent responses based on the dialogue context provided. To help you understand this task, we provide 1 example below. EXAMPLE 1: The example memory is: User: - Divorced - Raising one child - Immigrated from Britain last year - Metal worker Assistant: - Not married - Girlfriend has 2 kids - Works on mTurk, landscaping, sales, envelope stuffing, painting - Used to love winter, but has become intolerant of it - Works with a friend who owns "John of all trades" The example dialogue context is: User: Today was the hottest day I've ever experienced here in Florida! So the response to the user is: do you enjoy the heat more than the cold in Britain? The following is the case you need to test: The test memory is:[previous memory] The test dialogue context is:[dialog] So the response to the user is:
Output	[response]

Table B.16: The prompt for single model evaluation.

Prompt	You are an impartial judge. You will be shown a Conversation Context, Personality of Speakers and Assistant Response. #Fluency: Please evaluate whether the Assistant's response is natural, fluent, and similar to human communication, avoiding repetition and ensuring a diverse range of output. #Consistency: Please evaluate whether the Assistant's response is consistent with the information of persona list. Any deviation from the expected personality may indicate a lack of coherence. #Coherency: Please evaluate whether the Assistant's response maintains a coherent and logical flow of conversation based on the evolving context. A response with good context coherence can understand and respond appropriately to changes in conversation topics, providing smooth and sensible interactions. Conversation Context:[dialog] Personality:[persona] Assistant Response: [response] Begin your evaluation by providing a short explanation, then you must rate the Assistant Response on an integer score of 1 (very bad) to 100 (very good) by strictly following this format: [[score]].
Output	[output]

Table B.17: The prompt of pairwise model evaluation.

Prompt	Hi! We are a group of researchers working on Artificial Intelligence. In this task, we will ask you to help us rate the assistant's responses. In the area below, you will first read: 1. A conversation context comes from two speakers (the user and bot) 2. The personality of two speakers (the user and bot) extracted from past dialogs. 3. Two responses from AI systems. Your task is to decide which response is better. There are several dimensions that you can think along. Consider the following questions: 1. Is the response coherent? A response with good context coherence can understand and respond appropriately to changes in conversation topics, providing smooth and sensible interactions. 2. Is the response consistent? Evaluate whether the response is consistent with the information of persona list. Any deviation from the expected personality may indicate a lack of consistency. 3. Is the response natural and fluent? Please evaluate whether the response is natural, fluent, and similar to human communication, avoiding excessive repetition and ensuring a diverse range of output. Based on your aesthetics, which one do you prefer? For example, you might prefer one poem over another poem. Ultimately, you should decide which response is better based on your judgment and your own preference. There are four options for you to choose from: 1.Response 1 is better : If you think response 1 has an advantage, then choose this option. 2.Response 1 is slightly better : Response 1 is very marginally better than response 2 and the difference is small. 3.Response 2 is slightly better : Response 2 is very marginally better than response 1 and the difference is small. 4.Response 2 is better : If you think response 2 has an advantage, then choose this option. There are cases where the difference between the two responses is not clear. In this case, you can choose the second or the third option. However, in general, we ask you to choose those options as few as possible. response 1: [response1] response 2: [response2]
Output	[response]

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al., 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774 .

An, C., Gong, S., Zhong, M., Li, M., Zhang, J., Kong, L., Qiu, X., 2023. L-eval: Instituting standardized evaluation for long context language models. arXiv preprint arXiv:2307.11088 abs/2307.11088.

Bae, S., Kwak, D., Kang, S., Lee, M.Y., Kim, S., Jeong, Y., Kim, H., Lee, S.W., Park, W., Sung, N., 2022. Keep me updated! memory management in long-term conversations, in: Findings of the Conference on Empirical Methods in Natural Language Processing.

Beltagy, I., Peters, M.E., Cohan, A., 2020. Longformer: the long-document transformer. arXiv preprint arXiv:2004.05150 .

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T.J., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners. arXiv preprint ArXiv:2005.14165 .

Chen, N., Li, H., Huang, J., Wang, B., Li, J., 2024a. Compress to impress: Unleashing the potential of compressive memory in real-world long-term conversations. arXiv preprint arXiv:2402.11975 .

Chen, S., Wong, S., Chen, L., Tian, Y., 2023. Extending context window of large language models via positional interpolation. arXiv preprint arXiv:2306.15595 .

Chen, Y., Qian, S., Tang, H., Lai, X., Liu, Z., Han, S., Jia, J., 2024b. Longlora: Efficient fine-tuning of long-context large language models, in: the International Conference on Learning Representations.

Choi, E., On, K.W., Han, G., Kim, S., Nam, D.W., Jo, D., Rho, S.E., Kwon, T., Seo, M., 2023. Effortless integration of memory management into open-domain conversation systems. arXiv preprint arXiv:2305.13973 .

Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C., 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in Neural Information Processing Systems 35, 16344–16359.

Deriu, J., Rodrigo, Á., Otegi, A., Echegoyen, G., Rosset, S., Agirre, E., Cieliebak, M., 2019. Survey on evaluation methods for dialogue systems. Artificial Intelligence Review 54, 755 – 810.

Dubois, Y., Li, C.X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., Guestrin, C., Liang, P.S., Hashimoto, T.B., 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. Advances in Neural Information Processing Systems 36.

GLM, T., Zeng, A., Xu, B., Wang, B., Zhang, C., Yin, D., Rojas, D., Feng, G., Zhao, H., Lai, H., et al., 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. arXiv preprint arXiv:2406.12793 .

Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M., 2020. Retrieval augmented language model pre-training, in: the International Conference on Learning Representations.

Kann, K., Ebrahimi, A., Koh, J.J., Dudy, S., Roncone, A., 2022. Open-domain dialogue generation: What we can do, cannot do, and should do next, in: NLP4CONVAI.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t., 2020. Dense passage retrieval for open-domain question answering, in: the Conference on Empirical Methods in Natural Language Processing.

Lee, G., Hartmann, V., Park, J., Papailiopoulos, D., Lee, K., 2023. Prompted llms as chatbot modules for long open-domain conversation, in: Findings of the Annual Meeting of the Association for Computational Linguistics.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kuttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., Kiela, D., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks, in: the Annual Conference on Neural Information Processing Systems.

Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B., 2016. A diversity-promoting objective function for neural conversation models, in: Annual Conference of the North American Chapter of the Association for Computational Linguistics.

Li, J., Wang, M., Zheng, Z., Zhang, M., 2023. Loogle: Can long-context language models understand long contexts? arXiv preprint arXiv:2311.04939 .

Liu, C.W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., Pineau, J., 2016.

How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation, in: the Conference on Empirical Methods in Natural Language Processing.

Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P., 2023. Lost in the middle: How language models use long contexts, in: Transactions of the Association for Computational Linguistics.

Lu, J., An, S., Lin, M., Pergola, G., He, Y., Yin, D., Sun, X., Wu, Y., 2023a. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. arXiv preprint arXiv:2308.08239 .

Lu, Q., Qiu, B., Ding, L., Xie, L., Tao, D., 2023b. Error analysis prompting enables human-like translation evaluation in large language models: A case study on chatgpt. arXiv preprint .

Mazaré, P.E., Humeau, S., Raison, M., Bordes, A., 2018. Training millions of personalized dialogue agents, in: the Conference on Empirical Methods in Natural Language Processing.

MetaAI, 2024. Llama 3. URL: https://llama.meta.com/llama3/. accessed: 2024-08-26.

Pan, A., Shern, C.J., Zou, A., Li, N., Basart, S., Woodside, T., Ng, J., Zhang, H., Emmons, S., Hendrycks, D., 2023. Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark, in: the International Conference on Machine Learning.

Papineni, K., Roukos, S., Ward, T., Zhu, W.J., 2002. Bleu: A method for automatic evaluation of machine translation, in: the Annual Meeting of the Associa-

tion for Computational Linguistics. URL: https://doi.org/10.3115/1073083.1073135.

Peng, K., Ding, L., Zhong, Q., Shen, L., Liu, X., Zhang, M., Ouyang, Y., Tao, D., 2023. Towards making the most of chatgpt for machine translation. arXiv preprint arXiv:2303.13780 .

Pu, X., Gao, M., Wan, X., 2023. Summarization is (almost) dead. arXiv preprint arXiv:2309.09558 abs/2309.09558.

Research, F.A., 2019. Faiss. URL: https://ai.meta.com/tools/faiss/. accessed: 2024-08-26.

Robertson, S., Zaragoza, H., et al., 2009. the probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval .

Rubin, O., Berant, J., 2024. Long-range language modeling with self-retrieval, in: Transactions of the Association for Computational Linguistics.

Shuster, K., Xu, J., Komeili, M., Ju, D., Smith, E.M., Roller, S., Ung, M., Chen, M., Arora, K., Lane, J., Behrooz, M., Ngan, W., Poff, S., Goyal, N., Szlam, A., Boureau, Y.L., Kambadur, M., Weston, J., 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. arXiv preprint arXiv:2208.03188 .

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al., 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 .

Wu, H., Wang, W., Wan, Y., Jiao, W., Lyu, M., 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. arXiv preprint arXiv:2303.13648 .

Wu, Q., Lan, Z., Qian, K., Gu, J., Geramifard, A., Yu, Z., 2022. Memformer: A memory-augmented transformer for sequence modeling, in: Findings of the Annual Meeting of the Association for Computational Linguistics.

Xu, J., Szlam, A., Weston, J., 2022a. Beyond goldfish memory: Long-term open-domain conversation, in: the Annual Meeting of the Association for Computational Linguistics.

Xu, X., Gou, Z., Wu, W., Niu, Z.Y., Wu, H., Wang, H., Wang, S., 2022b. Long time no see! open-domain conversation with long-term persona memory, in: Findings of the Annual Meeting of the Association for Computational Linguistics.

Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., Tam, W.L., Ma, Z., Xue, Y., Zhai, J., Chen, W., Zhang, P., Dong, Y., Tang, J., 2022. Glm-130b: An open bilingual pre-trained model. arXiv preprint arXiv:2210.02414 .

Zhang, C., D'Haro, L.F., Chen, Y., Zhang, M., Li, H., 2023a. A comprehensive analysis of the effectiveness of large language models as automatic dialogue evaluators, in: AAAI Conference on Artificial Intelligence.

Zhang, P., Xiao, S., Liu, Z., Dou, Z., Nie, J.Y., 2023b. Retrieve anything to augment large language models. arXiv preprint arXiv:2310.07554 .

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J., 2018. Personalizing dialogue agents: I have a dog, do you have pets too?, in: the Annual Meeting of the Association for Computational Linguistics.

Zhang, T., Liu, Y., Li, B., Zeng, Z., Wang, P., You, Y., Miao, C., Cui, L., 2022. History-aware hierarchical transformer for multi-session open-domain dialogue system, in: Findings of the Conference on Empirical Methods in Natural Language Processing.

Zhong, Q., Ding, L., Liu, J., Du, B., Tao, D., 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. arXiv preprint arXiv:2302.10198 .

Zhong, W., Guo, L., Gao, Q., Ye, H., Wang, Y., 2024. Memorybank: Enhancing large language models with long-term memory, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 19724–19731.

Zhou, J., Chen, Z., Wang, B., Huang, M., 2023. Facilitating multi-turn emotional support conversation with positive emotion elicitation: A reinforcement learning approach, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1714–1729.