SummChat: Leveraging Virtual Context with Dual LLMs for Efficient Chat Tokenization

Anonymous ACL submission

Abstract

This paper introduces SummChat, a novel approach to enhance token efficiency in conversational agents via dual LLMs leveraging a virtual context. Focused on multi-round conversation situations, SummChat integrates a second and inexpensive LLM to act as a token reduction model between the user and the main language model. This model processes user prompts before reaching the main model, which allows the input to be reduced. This secondary model can efficiently eliminate extraneous information while providing sufficient context for the more advanced main model to answer appropriately. Additionally, this token-reduced prompt remains comprehensible to a human observer to facilitate greater downstream applications. This token-reduction method is enhanced by the use of virtual context, which is used to preserve original user prompts in conversational history, allowing the main model to retrieve specific user-provided information if needed. This system facilitates preservation of response quality across multi-round conversations.

Experimental results indicate an average response quality degradation of only 2.05% in exchange for a 13.26% reduction in input token usage when compared with SOTA. This results in an improvement of 12.4% in quality per 100 tokens. This paper demonstrates SummChat's potential in balancing response quality and cost-effectiveness, providing a new technique through which future works can leverage powerful LLMs in a more cost-efficient manner.

1 Introduction

Large Language Models (LLMs) such as GPT-4, Gemini, and LLaMA [Dee; Ope23; Tou+23] have significantly impacted the field of NLP. These powerful models are now widely used across various domains such as task planning [Yua+23; Lon+23], conversational agents [Pac+23; Abb+23; LYS23; Don+23], and recommendation systems [Gao+23; Fan+23; ASO23]. However, the widespread adoption of LLMs faces a significant hurdle: the complex infrastructure and resource requirements of advanced LLMs results in the need for a vast amount of computational resources or substantial API usage fees.

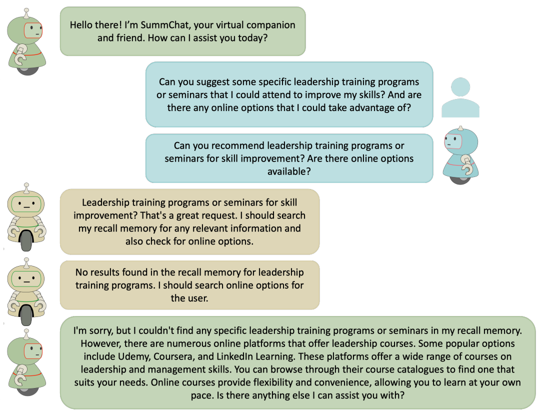

Figure 1: An example discussion using SummChat, displaying the SummChat dialogue (green), the user prompt and processed user prompt (blue), as well as the inner thought dialogue (creme)

Both of these are dependent on the number of input tokens to process. For example, OpenAI bases its API costs on the number of input and output tokens, making extended use financially prohibitive for many users and organisations. This is of key importance as the use cases of LLMs expand and require larger contexts, such as using LLMs to perform function calls and needing to explain these functions in the context window [Pac+23], or using LLMs for robot planning and navigation and thus including the various action or navigation priors in the system prompt [Lon+23; Hua+23]. Thus, reducing this issue of token usage has the potential to improve LLM accessibility across a wide range of increasingly complex applications, to individuals and institutions with lower economic capabilities. Additionally, while reducing token usage is pivotal for avoiding costs, it is also necessary to have reduced prompts preserve semantic information and remain comprehensible to a human observer. This has many benefits including the ability to be combined with human in-the-loop reinforcement learning, improving the transparency of medical chatbots, and improving human computer interaction aspects of conversational AI overall through additional clarity.

This paper introduces SummChat, a novel approach that directly addresses the challenge of token usage minimisation via a dual LLM system, coupled with a virtual context. SummChat proposes an input processing pipeline, as shown in figure 2 that integrates a cost-effective LLM within a conversational agent system. This secondary LLM analyses and summarises the user prompt before forwarding it to the main LLM embedded within the conversational agent. This process eliminates irrelevant segments of user prompts, simplifies phrasing, and retains key semantic information, leading to a significant reduction in token usage and, consequently, lower API costs.

While prioritising cost reduction, SummChat also manages to maintain response quality. This is done by leveraging conversational history, alongside a virtual context space with access to an external context space. This context space implements efficient information storage and retrieval, and can be accessed by the main language model through the use of function calls for independent action and information retrieval. SummChat additionally improves on this memory usage by altering the way in which information is stored in external context, and utilised by the cost-effective summary LLM. This combination ensures that SummChat delivers accurate and insightful responses, even within resource-constrained environments. An example of this summarisation procedure can be seen in figure 1 with the green colour showing SummChat's user-facing responses (provided by the main model), creme showing the internal thought process of SummChat (also provided by the main model), and blue showing the initial user prompt followed by the token reduced user prompt (provided by the token reduction model).

This paper demonstrates the effectiveness of SummChat with a focus on the domain of chat applications. This is to detail the effect of the token minimisation in a setting where it will be of most use, due to the variety of context lengths involved wtihin multi-round dialogues. In this setting, state-of-the-art methods like MemGPT[Pac+23] excel in response quality, but struggle with an accumulating high token usage over extended conversations. Additionally, other prompt compression methods, such as [Jia+23], present issues for human-computer interaction as their compressed prompts become incomprehensible to human users, hindering the valuable input of human feedback. In a conversational AI setting, users who see the summarised message as shown in figure 1 will be able to comprehend the summary and confirm that their meaning is effectively conveyed, thus making for a more seamless user experience.

The proposed method, SummChat, achieves a reduction in token usage of 13.26% while minimising change in response quality to -2.05%, offering a novel solution for cost-effective and efficient conversational AI that fosters a more engaging and accessible LLM usage experience across a diverse set of applications.

To summarise, the key contributions of this paper are as follows:

1. Provide a novel dual LLM pipeline equipped with virtual context for conversational memory
2. Leveraging conversational history summarisation to improve token reduction methods
3. Providing comprehensible token-reduced prompts, enabling human feedback and improved human-computer interaction
4. Provide a token usage reduction of 13.26% whilst only altering quality by -2.05% resulting in an improvement in quality per 100 tokens of 12.4% compared to the current SOTA

2 Background and Related Work

Large Language Models (LLMs). LLMs are an area of increasing development in recent years with a variety of larger and more intelligent models such as GPT-4, Gemini and LLaMA being released [Ope23; Dee; Tou+23]. The popularity of these works has also led to works that expose issues in current LLMs, and propose solutions for improving their capabilities. For example, a limited context window being fixed by virtual context systems [Pac+23], lack of visual grounding being fixed via multi-modal LLMs [Ope23] or visual-embeddings [Zhu+23; Maa+23], and lack of embodied grounding for robotic applications [Col+23; Dri+23]. This

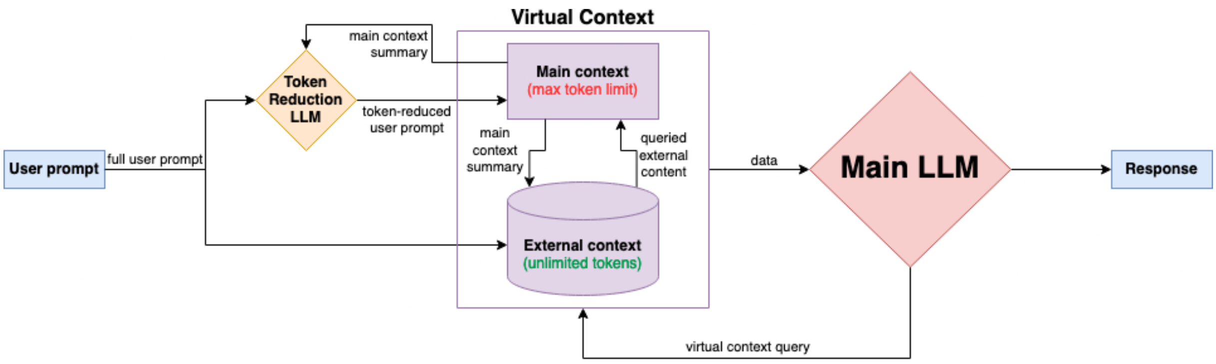

Figure 2: The SummChat pipeline incorporates a secondary language model coupled with an extensible virtual context which treats user prompts in two ways. The full user prompt is passed into a token reduction model, using a summary of the main context in order to effectively reduce the number of tokens consumed by the user prompt. The full user prompt is also stored in external context, eliminating information loss during the token reduction process. This allows for the retrieval of high-quality responses from the main model at a lower input token cost.

paper differs in the fact that, to our knowledge, it is the first to focus on the cost of token usage in a multi-round conversational setting, and the first to leverage virtual context to enhance the information available when using summarised prompts. This paper also focuses on the application of token reduction models to conversational settings where multi-round conversations occur, and where the utilisation of conversational history can significantly increase token usage over time. This allows the efficacy of the proposed method to be demonstrated on context windows of various sizes.

Reduction of Token Usage in LLMs. Competing works in this field focus on the reduction in length of data called from databases [Liu+23] or on prompt compression to reduce token usage [Jia+23]. These works differ from our approach in two significant ways: usage of fine-tuned prompt reduction models, or compression of the prompt into a form incomprehensible to a human observer. The proposed approach instead uses an off-the-shelf LLM to reduce and summarise prompts, with an additional focus on semantic sense of the reduced prompt to a human observer. Furthermore, our approach features metrics based on direct token usage, alongside effective summarisation of user prompts and conversation history, all while preserving response quality for conversational agents.

3 Method

3.1 Overview

The core of the proposed approach lies in how the user prompt is treated before it is passed on to the main language model. SummChat introduces a novel input processing pipeline that specifically targets token reduction. This pipeline has two major constituent components. The first is the summarisation and token reduction model which acts upon the user prompt prior to being input into the main model. The second component is our context handling pipeline, which allows the main model to retrieve information removed during the token reduction process, if needed.

3.2 Token Reduction LLM

The token reduction model functions by rewriting and editing user prompts, eliminating superfluous token usage by removing extraneous information and words. To this end, SummChat implements this secondary model in the processing pipeline using a smaller scale LLM with reduced computational and API costs. Older and less computationally intensive models are utilised for this secondary model in order to ensure that the additional cost of computing a token-reduced version of the user prompt does not offset the savings acquired by inputting fewer tokens into the more advanced main model.

In order to ensure that the secondary LLM efficiently and predictably summarises and reduces the token usage of user prompts, a purpose-made system prompt is used. This prompt is composed of two constituent halves. The "base" prompt contains basic instructions for the secondary model that dictate its task and the constraints under which the token-reduced user prompt must be produced. This base prompt is unchanged during conversation with the SummChat agent. The second half of the prompt contains a summarised version of the current conversation and is dynamically updated

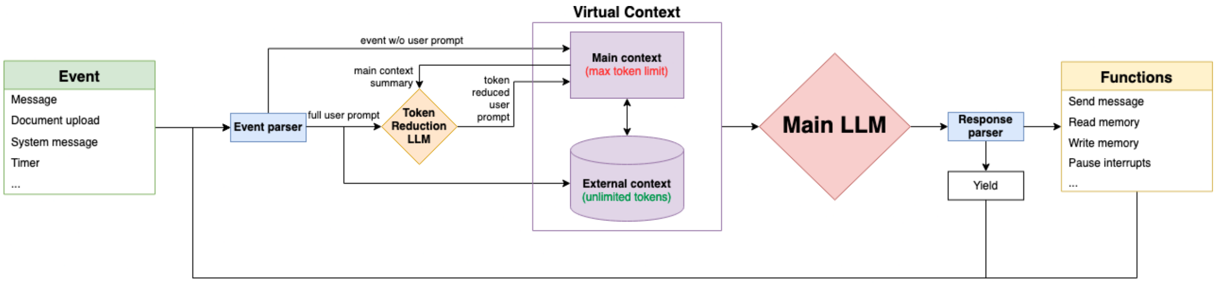

Figure 3: The proposed input processing pipeline is then embedded in a larger conversational agent structure. The event parser functions as in MemGPT for non-user-driven events like timer events and system messages while incorporating special handling for user prompts. The effect of this change is that SummChat keeps the event driven nature of the pre-existing approach, while enabling the reduction of token usage when passing user prompts into the main language model.

every time the token-reduction model is prompted. The model is informed in the prompt that it may use the conversation summary to inform itself of what aspects of the user prompt are most conversationally relevant. This is provided in order to facilitate the model removing or summarising information not critical to the conversational history when formulating a response to the user prompt.

As a whole, the token reduction model and the dynamic prompt provide an intermediate processed input that retains key conversationally relevant information in user prompts at a lower token usage. The purpose-made prompt is effective enough to provide this result consistently. Additionally, by instructing the model to provide a human-comprehensible output in the system prompt, the proposed method also ensures that the token-reduced output is comprehensible to human observers and not just the main language model.

3.3 Context and Memory Usage

Rephrasing, summarisation, and rewriting of user prompts yields token savings, but it inevitably also introduces information loss. To address this issue, SummChat employs a virtual context system.

This is achieved by providing the LLM with a programmatic interface to a separate storage system, where information beyond the primary context window can be stored and accessed. Function calls then act as a bridge between these two systems, seamlessly transferring data back and forth, just as in [Pac+23]. Through these calls, the model can search for specific data, inject new knowledge, and even update existing information, dynamically expanding its understanding during the course of a conversation.

The proposed input processing pipeline makes use of virtual context by directly and automatically storing the original, unadulterated user prompt into external context. This provides the main LLM with the ability to query the external context for specific information provided in user prompts if it is evaluated to be relevant to facilitate responding to user prompts. Information retrieval from external context is still efficient in terms of input tokens; the external context can retrieve information that is specifically relevant to the main model's query, instead of simply providing a copy of the entire user prompt. In the case of SummChat, we leverage a local external context implementation as in [Pac+23] which splits user prompts into passages before storing them as text embeddings, allowing for efficient retrieval of archived information.

The end result of this approach is that the downsides of the token reduction model are greatly mitigated, even in cases where information is omitted during the token reduction process. Coupled with an efficient external memory information retrieval method, token savings can be further preserved by only sending information relevant to the main model's query when information omitted during the token reduction process is required.

3.4 Conversational agent implementation

The proposed method provides a fully interactive chat system by implementing the input processing pipeline into the existing framework of a conversational agent, such as MemGPT[Pac+23].

Several key components from the conversational agent implementation of MemGPT are further implemented in the proposed method. One of these components is the event system. Certain events within the conversation or from the external world can trigger automatic updates to the main context window, bringing relevant information from the external memory into focus. These events can be driven by user interactions or the system. Using this event system allows the main language model to independently query for information and send system messages in an automated manner without user intervention. An event parser then handles the specific characteristics of these events, sending relevant information to the language model and extending the conversational history.

Due to the fact that, in this design, user prompting is treated as an event, SummChat modifies the conversational agent's event parser and implements differentiated behaviour based on event type. The SummChat event parser retains existing behaviour for non-user-driven events such as timer events and system messages, while incorporating a novel handling method for user prompts; this enables the reduction of token usage when passing the user's messages into the main language model.

The proposed approach's handling of the constituent parts of the virtual context also implements new interaction flows with MemGPT's implementation of a conversational agent's virtual context. The latter naively feeds all new events into the main context for the main language model to handle. The proposed approach only does this for non-user events. As in the proposed pipeline, the conversational agent implementation feeds the token-reduced user prompt into main context while simultaneously feeding the unadulterated user prompt into external context for later retrieval. Additionally, we directly pull a summary of the main context to provide this as part of the token-reducing model system prompt. However, we do retain the existing behaviour of the conversational agent that summarises the main context and stores the summary in the external context once the main context reaches its token limit.

Beyond the virtual context stage of the pipeline, all implemented components in the conversational agent are retained. This includes the system prompts for the main language model, the response parsers, and function call implementations.

This design allows the proposed approach to integrate the benefits of this conversational agent, including the infrastructure for accessing virtual context and the event-driven nature of this approach.

4 Experiments and Evaluation

4.1 Primary Experiments

The proposed approach is evaluated based on the following criteria:

1. Does the approach provide responses that are of a comparable quality to the SOTA?

2. Does the approach yield an appreciable reduction in input token usage numbers?

To test these criteria, the Ultrachat_200k dataset is used. Ultrachat_200k is a dataset comprised of a filtered version of the Ultrachat dataset used to train the Zephyr-7B model[Tun+23]. This dataset contains chat logs generated by conversations held between a ChatGPT agent acting as a user and another ChatGPT agent acting as an assistant. Each log is composed of an initial user prompt followed by a reply from the assistant and an ensuing back-and-forth conversation between the user and the assistant. The length of these logs is variable, with some conversations containing as few as 6 messages (3 conversation rounds) and others being significantly longer.

For the main evaluation, 100 samples were selected at random from the ultrachat_200k dataset. During the experiments, we collected the responses generated by SummChat (the proposed approach), as well as the number of input tokens used. Additionally, we collected the same data for MemGPT, which served as the SOTA comparison.

MemGPT is selected as the point of comparison as it has demonstrated SOTA performance in conversational settings, particularly over extended conversations. These are the situations where input token use can accumulate most severely due to the build up of conversational history each round. Additionally, MemGPT does not process the user prompt prior to the main LLM responding to the user, thus creating a useful frame of reference for the proposed pipeline. Furthermore, we can directly compare how MemGPT's implementation of an event-driven conversational agent fares in response quality relative to token use, compared to the proposed implementation which leverages several of MemGPT's components.

Using this data, the proposed approach and baseline were evaluated on three key metrics:

1. GPT-4 Evaluation (GPT-4 Eval), used to measure overall response quality

2. Token Usage, used to measure cost and computational requirements

3. GPT-4 Evaluation per 100 tokens (Eval per 100 tokens), used to get a holistic measure of efficiency based on quality and cost

GPT-4 Eval is used as a quality metric, as on evaluation tasks, this model's responses were shown to align highly with the judgements of human experts [Zhe+23]. To calculate GPT-4 Eval, GPT-4 was provided with the responses from both SummChat and MemGPT; alongside these, GPT-4 was also provided with the user prompt and the ground-truth response from the dataset for each conversation round in order to provide some context during the evaluation. GPT-4 was then asked to provide a score between 0-100 for both SummChat and MemGPT's responses, with higher scores indicating a better response.

The acquired token usage and response quality numbers are presented in table 1 with percentage differences between the proposed method and SOTA being shown in table 2. SummChat displayed a significant 13.26% reduction in input token usage in exchange for only a 2.05% degradation in GPT-4 Eval. This results in a 12.40% improvement in Eval per 100 tokens when compared to MemGPT. Additionally, experiment results showed that SummChat yielded equivalent or higher response scores than MemGPT in 52.32% of conversation rounds as shown in figure 3 and that the token saving increases in longer multi-round conversations, shown in figure 4.

Table 2: Percentage Change Metrics

Metric	Score
Eval Change	-2.05%
Token Saving	**13.26%**
Eval/Token Change	**12.40%**
Percentage Favoured	**52.32%**

4.2 Ablation Study

To evaluate the impact of each of the contributions in the proposed pipeline, an ablation study was conducted. The novel components of the proposed input processing pipeline were ablated. Specifically, the proposed implementation of SummChat was compared to versions of SummChat that had:

1. The summary removed from the system prompt provided to the token reduction model (SummChatxSummary)

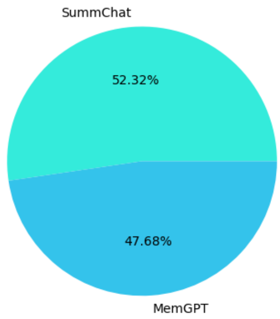

Figure 4: This graph shows the percentage of times SummChat performed equal to or better than MemGPT in GPT-4 Evaluation Score

2. The automatic upload of full user prompts into the external context disabled (SummChatx-Context)

3. Both of the prior ablations applied together (SummChatxBoth)

Evaluation metrics were gathered in accordance with the primary evaluation utilising the same metrics and dataset. However, 25 randomised samples were used, ensuring 15 of these samples contained extended long-form conversations (>4 question-answer rounds) while 10 were from short-form conversations. This was done in order to ensure the evaluation accurately represented the performance of each agent across both short and long conversation samples. The result of the ablations on this dataset can be seen in table 3 with the percentage difference range between the proposed method and ablations shown in table 4

SummChat without any ablations. The proposed SummChat implementation performs demonstrably better overall than the ablated versions. It displays the highest GPT-4 Eval score and, despite having the highest token use overall, also shows the highest Eval per 100 tokens.

SummChat without the summary provided in the token reduction model's system prompt. When compared to Summchat, it can be reasoned that the reduced contextual information leads to the token-reducing model removing information that is relevant to the conversation at hand, consequently reducing the response quality. We can see this from

Table 1: Evaluation of Key Performance Metrics for SummChat and MemGPT

Model	GPT-4 Eval	Token Use	Eval per 100 Tokens
SummChat (**ours**)	84.09	**2942.99**	**2.90**
MemGPT	**85.85**	3392.70	2.58

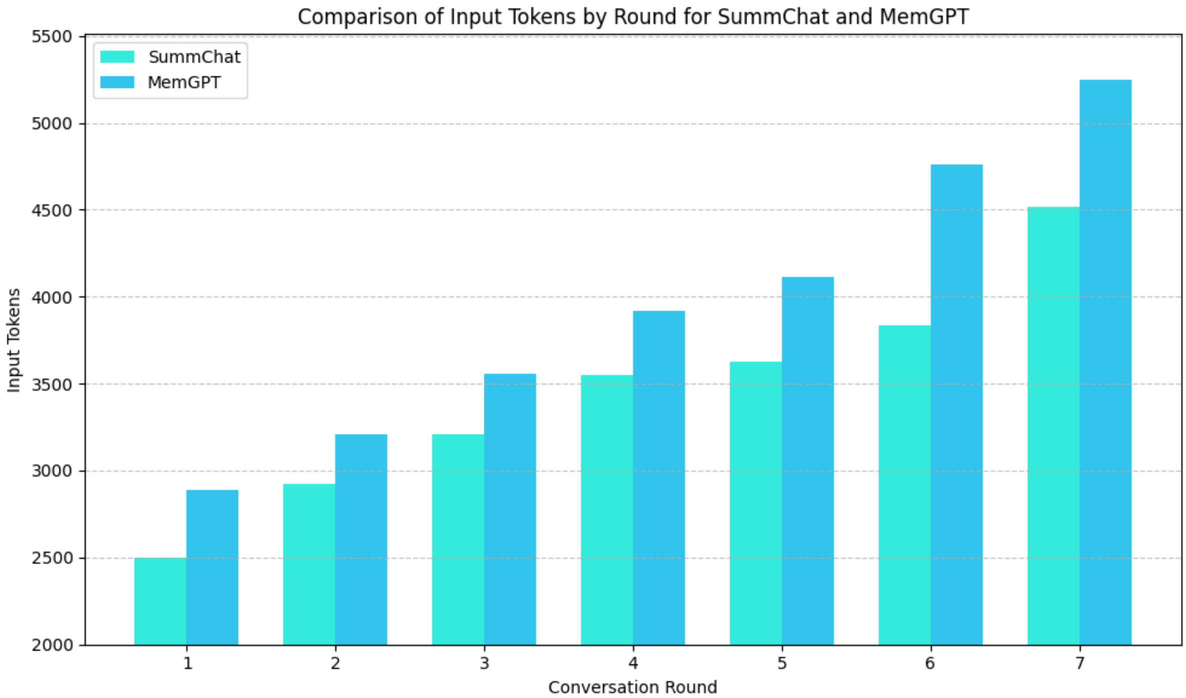

Figure 5: This graph shows the tokens used per round of conversation for both SummChat and MemGPT

the fact that the ablated agent displays both significantly lower response scores and somewhat lower token usage. Overly-summarised user prompts, or erroneously summarised user prompts, will logically lead to poor response quality. The worsened prompt summarisations can have a cumulative effect, causing the main language model to provide poorer-quality future responses. Hence, the ablation here suggests that the conversational summary helps the token reduction model more effectively summarise user prompts. Additionally, worsened knowledge of the information provided by the user has the potential to impact the main model's ability to effectively query external memory for information in previous user prompts.

SummChat without full user prompt upload into external context. A noticeable degradation in GPT-4 Eval can be seen when compared to the proposed implementation of SummChat. This is due to the agent being unable to search external context for details provided in user prompts. This presents a challenge for conversations with long-form user queries and conversations. However, as this ablated agent still contains the summary within the token reduction model's system prompt, the token reduction agent is able to effectively summarise user prompts, thus enabling the main model to provide responses of only marginally diminished quality. The slightly lower token usage of this ablated version compared to the proposed SummChat can likely be explained by the fact that, when querying for additional information in the external context, the virtual context will send no data in return. This would not have been the case had the full user prompts been input into the external context; in this case, the main model would have received a response with data, thereby driving up the input token count, particularly in longer conversation samples, which represent the majority of our ablation study dataset.

SummChat with a full ablation. The final ablation presents a GPT-4 Eval score and token use result, which are favourable in quality to the other two ablations but still fail to reach the quality of the implemented SummChat pipeline. There are two likely reasons for this. First, response qual-

Table 3: Evaluation of Performance Metrics on Ablation Studies

Model	GPT-4 Eval	Token Use	Eval per 100 Tokens
SummChat	**85.14**	3044.19	**2.86**
SummChatxSummary	75.88	**2885.38**	2.58
SummChatxContext	82.86	2981.12	2.82
SummChatxBoth	82.82	2981.96	2.83

Table 4: Percentage Change Metrics on Ablation Studies

Metric	Lowest Diff	Highest Diff
GPT-4 Eval	**2.68%**	**10.88%**
Token Use	-2.04%	-5.22%
Eval per 100 Tokens	**1.10%**	**7.85%**

ity: due to GPT-4's context size of 8000 tokens, it is possible for the main model to capture the full conversation of several data samples. It can still be seen that the proposed SummChat provides a higher average evaluation score, thus, suggesting that in dataset samples with the longest conversations, the implemented version of SummChat is able to get higher quality responses due to its access to full user prompts. The token usage count is explained by the virtual context's lack of available information during external context queries, as discussed in the exploration of SummChat without full user prompt upload into the external context.

5 Conclusion

This research demonstrates the effectiveness of SummChat, a novel dual LLM and virtual context architecture, in significantly reducing input token usage in conversational AI systems while maintaining high response quality. SummChat achieves a 13.26% decrease in token usage compared to existing state-of-the-art models, with a minimal decline in response quality of only 2.05%. This translates to a notable 12.4% improvement in quality per 100 tokens used, representing a substantial gain in conversational agent efficiency. These findings highlight the ability of SummChat to balance cost and performance considerations effectively. By reducing token usage, SummChat paves the way for increased accessibility and affordability of LLMs for conversational AI applications. This, in turn, has the potential to broaden LLM adoption and facilitate the development of more engaging and accessible conversational AI experiences across diverse domains. Furthermore, etaining comprehensibility in the shortened prompt unlocks additional uses due to its advantages for human-computer interaction. This translates to, among others, more seamless user experiences in conversational AI systems.

6 Limitations and Future Work

Accuracy of summarisation in long user prompts. The effectiveness of SummChat relies on the accuracy of the token reduction LLM's summarisation. If the summarisation is inaccurate or omits crucial information, it could lead to the main LLM generating incorrect or incomplete responses. Poor summarisations are more common in user prompts with large bodies of texts, and where user requests reference specific parts of said text. This is greatly diminished by the availability of the full user prompt in external context, but the main language model may not always choose to query external context before responding to the user prompt. Fine-tuning the token-reduction model for this task may yield even greater response quality in future work.

Storage consumption of full user prompts stored in external context. Consistently storing the entirety of user's prompts in external context has storage cost implications. However, the current cost trade-off between storage and token use heavily favours the proposed approach. There are potential ways to mitigate this issue; in the case of SummChat, storing user prompts as embeddings helped reduce storage consumption, for instance. However, storage consumption is likely to be a less avoidable issue when dealing with large numbers of users, and as the conversational agent is used over significantly extended periods of time. We leave further exploration of this issue for future work.

7 References

References

[Abb+23] Mahyar Abbasian et al. *Conversational Health Agents: A Personalized LLM-Powered Agent Framework*. Dec. 2023. arXiv: 2310.02374 [cs]. (Visited on 12/15/2023).

[ASO23] Arkadeep Acharya, Brijraj Singh, and Naoyuki Onoe. "LLM Based Generation of Item-Description for Recommendation System". In: *Proceedings of the 17th ACM Conference on Recommender Systems*. RecSys '23. New York, NY, USA: Association for Computing Machinery, Sept. 2023, pp. 1204–1207. ISBN: 9798400702419. DOI: 10.1145/3604915.3610647. (Visited on 12/15/2023).

[Col+23] Open X.-Embodiment Collaboration et al. *Open X-Embodiment: Robotic Learning Datasets and RT-X Models*. Oct. 2023. arXiv: 2310.08864 [cs]. (Visited on 11/12/2023).

[Don+23] Xin Luna Dong et al. "Towards Next-Generation Intelligent Assistants Leveraging LLM Techniques". In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '23. New York, NY, USA: Association for Computing Machinery, Aug. 2023, pp. 5792–5793. ISBN: 9798400701030. DOI: 10.1145/3580305.3599572. (Visited on 12/15/2023).

[Dri+23] Danny Driess et al. *PaLM-E: An Embodied Multimodal Language Model*. Mar. 2023. arXiv: 2303.03378 [cs]. (Visited on 11/12/2023).

[Fan+23] Wenqi Fan et al. *Recommender Systems in the Era of Large Language Models (LLMs)*. Aug. 2023. arXiv: 2307.02046 [cs]. (Visited on 12/15/2023).

[Gao+23] Yunfan Gao et al. *Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System*. Apr. 2023. arXiv: 2303.14524 [cs]. (Visited on 12/15/2023).

[Hua+23] Siyuan Huang et al. *Instruct2Act: Mapping Multi-modality Instructions to Robotic Actions with Large Language Model*. May 2023. arXiv: 2305.11176 [cs]. (Visited on 06/05/2023).

[Jia+23] Huiqiang Jiang et al. *LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models*. Dec. 2023. arXiv: 2310.05736 [cs]. (Visited on 12/15/2023).

[LYS23] Lizi Liao, Grace Hui Yang, and Chirag Shah. "Proactive Conversational Agents in the Post-ChatGPT World". In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '23. New York, NY, USA: Association for Computing Machinery, July 2023, pp. 3452–3455. ISBN: 978-1-4503-9408-6. DOI: 10.1145/3539618.3594250. (Visited on 12/15/2023).

[Liu+23] Junyi Liu et al. *TCRA-LLM: Token Compression Retrieval Augmented Large Language Model for Inference Cost Reduction*. Oct. 2023. arXiv: 2310.15556 [cs]. (Visited on 12/15/2023).

[Lon+23] Yuxing Long et al. *Discuss Before Moving: Visual Language Navigation via Multi-expert Discussions*. Sept. 2023. arXiv: 2309.11382 [cs]. (Visited on 10/19/2023).

[Maa+23] Muhammad Maaz et al. *Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models*. June 2023. arXiv: 2306.05424 [cs]. (Visited on 06/27/2023).

[Ope23] OpenAI. *GPT-4 Technical Report*. Mar. 2023. arXiv: 2303.08774 [cs]. (Visited on 12/15/2023).

[Pac+23] Charles Packer et al. "MemGPT: Towards LLMs as Operating Systems". In: *arXiv preprint arXiv:2310.08560* (2023).

[Tou+23] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. July 2023. arXiv: 2307.09288 [cs]. (Visited on 11/12/2023).

[Tun+23] Lewis Tunstall et al. *Zephyr: Direct Distillation of LM Alignment*. Oct. 2023. arXiv: 2310.16944 [cs]. (Visited on 12/15/2023).

[Yua+23] Haoqi Yuan et al. *Plan4MC: Skill Reinforcement Learning and Planning for Open-World Minecraft Tasks*. Mar. 2023. arXiv: 2303.16563 [cs]. (Visited on 06/05/2023).

[Zhe+23] Lianmin Zheng et al. "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena". In: *ArXiv* abs/2306.05685 (2023). URL: https://api.semanticscholar.org/CorpusID:259129398.

[Zhu+23] Deyao Zhu et al. *MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models*. Apr. 2023. arXiv: 2304.10592 [cs]. (Visited on 06/05/2023).

[Dee] Google Deepmind. *Gemini - Google DeepMind*. https://deepmind.google/technologies/gemini/. (Visited on 12/15/2023).