A Prefetch-based LLM System Using Response Reuse*

Yeon-Ju Han, So-Hyun Park†, and Il-Gu Lee*
Department of Convergence Security Engineering, Sungshin Women's University, Korea
{220256182, sohyunpark, iglee}@sungshin.ac.kr

Abstract

This study introduces a novel, prefetch-based inference methodology designed to address the critical challenge of computational inefficiency in large language models (LLMs). Recognizing the high costs associated with redundant computations, this method aims to optimize resource utilization by intelligently reusing past interactions. When a new query is received, the system proactively searches a repository for semantically similar past query-response pairs. This identification is achieved based on high-dimensional embedding similarity, allowing the model to find contextually relevant precedents even if the phrasing is not identical. Once a suitable match is found, the method utilizes this historical information to systematically reconstruct a candidate response for the new query. A crucial validation phase is integrated into this process. To ensure the fidelity and reliability of the output, the candidate's response undergoes a rigorous verification process to check its quantitative and logical consistency. This step is essential for filtering out potential inaccuracies that may arise from adapting previous, non-identical answers. By strategically intercepting and resolving queries without invoking a full, costly inference cycle, the proposed method substantially minimizes unnecessary LLM calls. This research demonstrates the significant potential of a prefetch-and-verify architecture, offering a practical pathway for designing more efficient, scalable, and cost-effective LLM inference mechanisms.

1 Introduction

Large language models (LLMs) have demonstrated superior performance in various natural language processing (NLP) tasks, such as question answering and document summarization, based on their ability to understand context and generate natural language [1]. However, generating responses to user queries using LLMs requires significant computation, raising concerns about operational costs and environmental impact [2]. In particular, re-executing the LLM for complex or repetitive queries results in resource waste and can lead to response latency, thereby degrading the user experience [3]. To address these issues, methods for efficient memory usage, such as computational optimization, and response reuse/semantic caching have been proposed [4, 5]. However, while computational optimization methods can reduce memory waste and speed up operations, they still have the limitation of re-executing the entire LLM inference process from scratch, even for similar queries. In the case of response reuse and semantic caching, conventional studies typically follow a binary approach, either perfect hit or complete discard. This creates inefficiency, as the existing response cannot be reused if the similarity falls below a threshold, forcing the LLM to run again. To solve this problem, this paper

* Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. W-5, December 16-18, 2025, Sapporo, Japan. © The copyright of this paper remains with the author(s).}

† Corresponding author

proposes a prefetch-based inference algorithm centered on reusing previous conversational responses. While this algorithm also discards responses based on a similarity threshold, it goes beyond the complete discard method. It aims to improve inference speed and reduce resource consumption by providing a similar existing response to the LLM as a reference.
Furthermore, the accuracy of the reused response is verified to enhance the quality of the response. This approach enables more efficient LLM utilization by maintaining response quality while reducing the number of LLM calls.

The contributions of this study are as follows.

- Propose a prefetch-based inference algorithm that enhances the computational efficiency of LLMs by reusing semantically similar past responses instead of performing redundant full-scale inference.
- Introduce a multi-stage similarity evaluation mechanism that combines keyword, semantic, and cosine similarities to improve the reliability of response retrieval and reuse decisions.
- Design a response verification process that quantitatively and logically validates the consistency of reused responses, ensuring both efficiency and answer quality.

The remainder of this paper is organized as follows. Section 2 reviews prior research on computation optimization and response reuse techniques for LLM inference. Section 3 presents the proposed prefetch-based inference algorithm, detailing its three main stages: similarity evaluation, adaptation, and LLM response generation. Section 4 concludes the paper and discusses potential applications and future research directions.

2 Related Work

Various studies have been conducted to efficiently enhance the inference process of Large Language Models in order to address their high computational costs and response latency issues. These studies can be broadly classified into computational optimization and response reuse/semantic caching.

2.1 Computational Optimization

This approach focuses on increasing the fundamental (low-level) speed of LLM inference. It aims to maximize computational efficiency when the LLM is called. This field concentrates on optimizing the attention operation, a core element of LLMs, and its memory usage. The Key-Value (KV) cache, generated for attention operations during LLM inference, occupies significant memory resources. This approach focuses on improving the fundamental (low-level) inference speed of the LLM. That is, it emphasizes maximizing internal operational efficiency when the LLM is called. In particular, research has focused on the attention operation, a key component of LLMs, and the memory efficiency of the Key-Value (KV) cache generated during this process. Conventional methods managed the KV cache inefficiently, leading to memory waste and fragmentation problems [4]. vLLM [3] solved these problems through Paged Attention technology. Paged Attention divides the KV cache into blocks, allocating and reusing memory only as needed, thereby reducing memory waste and dramatically improving overall throughput for batch processing. These computational optimization techniques have significantly improved the inference efficiency of LLMs. However, these approaches focus only on how to finish the computation faster once a query is received. Consequently, even when a query that is semantically similar to one answered in the past is received, they cannot avoid the fundamental inefficiency of having to re-execute the entire LLM inference process from the beginning.

2.2 Response Reuse and Semantic Caching

To overcome the limitation of this redundant computation, the focus shifts to Response Reuse and Caching, which aims to eliminate unnecessary LLM calls themselves. Initial caching methods only returned a response if the prompt matched exactly at the string level, but this had very low utility. To improve this, semantic caching techniques such as GPTCache [5] have been proposed. This method stores and utilizes queries in the cache based on semantic similarity, not just identical matches, when a new query is input. This approach presented a practical way to eliminate LLM calls by returning existing responses based on embedding similarity. However, relying solely on embedding similarity can lead to limitations in precise semantic discrimination. As pointed out in the Efficient Prompt Caching [3] study, a semantically similar query does not guarantee that it can be answered with the same response. This makes it difficult to reuse the response as-is, even with high similarity. Conventional semantic caching follows only a binary "perfect hit" or "complete discard" approach. This causes significant inefficiency: if the similarity falls below a threshold, the entire cached response cannot be reused, and the LLM must be run again from the beginning. Therefore, to overcome these limitations, this paper proposes a recycling-based prefetch inference algorithm that utilizes the cached response as a reference for supplementary computation by the LLM, instead of discarding it.

3 Proposed Method

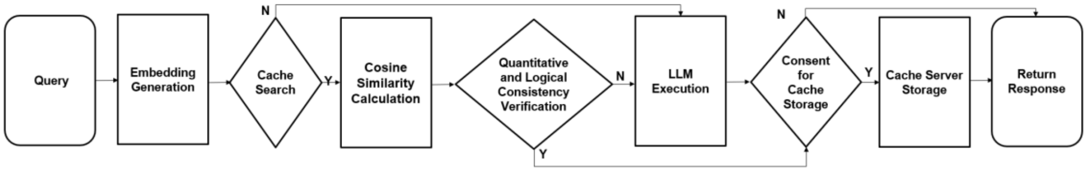
Query
Embedding Generation
N
Cache Search
Y
Cosine Similarity Calculation
Quantitative and Logical Consistency Verification
N
Y
LLM Execution
N
Consent for Cache Storage
Y
Cache Server Storage
Return Response

Figure 1. Block Diagram of LLM-based Query Response Reuse

As shown in Figure 1, the proposed method comprises three stages: similarity evaluation, adaptation, and LLM response generation. The similarity evaluation is the stage where an embedding vector is generated for a new user query, and its similarity is compared with the vectors of existing query-response pairs stored in the cache. During the conversion process to an embedding vector, synonyms and similar words can be automatically identified. The embedding model converts text into vectors based on its pre-trained information. Consequently, identical words are converted to identical vectors, and similar words are converted to similar vectors. If the embedding model receives a new word it has not learned, the model breaks the word down into subwords, finds the embeddings for each subword, and combines these embeddings to convert it into a new vector. Afterward, it retrieves only the query vectors from the query-response pairs stored in the cache and checks the similarity against all query vectors. Similarity is calculated using keyword, semantic, and cosine similarity. Keyword similarity measures the number of identical words as the cached query, and semantic measurement identifies semantic similarity through a thesaurus. Cosine similarity calculates similarity by measuring the angle between vectors. Subsequently, the results of the three similarity calculations are weighted and averaged to complete the final similarity calculation, and the query vector with the highest similarity is passed to the adaptation stage. In the adaptation stage, the reuse status and scope of the response are determined based on the calculated similarity score, and it is verified whether the selected response is actually reusable. That is, the reuse status and scope are determined based on the two vectors: the cached query vector received from the similarity evaluation stage and the input query vector. The reuse status is classified as 'full reuse,' 'partial reuse,' or 'no reuse.' In the case of full reuse, if the two vectors exceed a certain score, the entire response value of the cached query is retrieved and determined as the scope of reuse. In the case of partial reuse, if the score falls below a certain threshold, the response value is partially reused. At this time, the vector is not converted back to text; instead, the entire cached response value is passed to the LLM. In the case of no reuse, if the similarity score is below 50%, the scope is defined as not using the cached response value at all. A prompt containing only the new query is constructed, and the LLM is executed anew to perform an independent inference operation. The verification process checks the accuracy of quantitative information and the logical consistency between sentences and responses, discarding those with low reliability. In the LLM response generation stage, if the cached response cannot be fully reused, a supplementary operation is performed through the LLM inference process to generate the final response. Supplementary operation refers to the case selected for 'partial reuse' in the Adaptation stage. When selected for partial reuse, the entire cached response value is provided to the LLM, and the LLM can use this as a reference during its inference process. The generated response is, upon the user's choice, stored in the cache server so it can be managed and recycled when similar queries are input in the future. Through this structure, the proposed algorithm can minimize unnecessary model calls while securing response consistency, and consequently, it can improve the LLM's operational efficiency and generally provide high-quality responses by utilizing cosine similarity.

4 Conclusion

To address the pressing challenges of high operational costs and significant response latency inherent in LLM, this study introduces a novel, prefetch-based LLM processing method. This method is strategically centered on the intelligent reuse of previous conversational responses. At its core, a new incoming query is systematically evaluated for embedding similarity against a repository of cached query-response pairs. This evaluation serves to determine if an existing response can be effectively reused. The system employs an adaptive reuse strategy. If the similarity is determined to be sufficiently high (i.e., above a high-confidence threshold), the cached response is returned immediately, by passing the need for an LLM call entirely. In cases where supplementation is necessary (i.e., the match is partial but relevant), the LLM is executed in a supplementary capacity to augment or refine only the required portion of the answer, rather than generating a new one from scratch. This granular approach minimizes the computational burden. This dual-pronged strategy—combining immediate reuse with partial supplementation—substantially curtails the total number of LLM calls and effectively alleviates response latency, thereby enabling highly efficient inference operations. The proposed method demonstrates high potential for practical application, particularly in knowledge-intensive domains such as law and medicine, which critically depend on the immediate and accurate utilization of vast specialized knowledge. Going forward, the practical viability and performance gains of this architecture will be rigorously strengthened and validated through comprehensive experimental verification.

Acknowledgments. This work is supported by the Ministry of Trade, Industry and Energy (MOTIE)under Training Industrial Security Specialist for High-Tech Industry [grant number RS-2024-00415520]supervised by the Korea Institute for Advancement of Technology (KIAT), the Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program [grant numberIITP-2022-RS-2022-00156310] and National Research Foundation of Korea (NRF) grant [RS-2025-00518150], and the Information Security Core Technology Development program [grant number RS-2024-00437252] supervised by the Institute of Information & Communication Technology Planning &Evaluation (IITP).

References

[1] OpenAI. (2023, March). GPT-4 Technical Report. Retrieved from https://arxiv.org/abs/2303.08774

[2] Gill, W., Elidrisi, M., Kalapatapu, P., Ahmed, A., Anwar, A., & Gulzar, M. A. (2024, March). MeanCache: User-centric semantic cache for large language model based web services. Retrieved from https://arxiv.org/abs/2403.02694

[3] Zhu, H., Zhu, B., & Jiao, J. (2024, February). Efficient Prompt Caching via Embedding Similarity. Retrieved from https://arxiv.org/abs/2402.01173

[4] Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., & Stoica, I. (2023, September). Efficient Memory Management for Large Language Model Serving with PagedAttention. Retrieved from https://arxiv.org/abs/2309.06180

[5] Fu, B., & Feng, D. (2023, December). GPTCache: An Open-Source Semantic Cache for LLM Applications. Retrieved from https://aclanthology.org/2023.nlposs-1.24.pdf

Faster Answers and Cost Savings. Retrieved from