# Design and Verification of a Query-Aware External Memory Pipeline for Small Language Models

Seowan Kim
*Department of Healthcare IT*
*Inje University*
Republic of Korea
gonstoro0521@oasis.inje.ac.kr

Sungmoon Park
*Department of Healthcare IT*
*Inje University*
Republic of Korea
qjatjr9958@oasis.inje.ac.kr

Hwansuk Choi
*College of AI·Software Convergence*
*Kyungnam University*
Republic of Korea
chs2024@kyungnam.ac.kr

Namhyun Yoo
*Department of Computer Engineering*
*Kyungnam University*
Republic of Korea
hyun43@kyungnam.ac.kr

Jinhong Yang*
*Department of Healthcare IT*
*Inje University*
Republic of Korea
jinhong@inje.ac.kr

***Abstract*—While the utilization of Small Language Models (sLMs) has recently expanded, the limited context window of approximately 2,048 tokens serves as a critical bottleneck in long-term memory tasks. Conventional truncation or simple summarization techniques lead to information loss and distortion, causing the "catastrophic forgetting" problem. This study proposes a "Query-Aware External Memory Pipeline" inspired by the human "Selective Attention" mechanism to address this issue. This methodology combines importance extraction with embedding-based filtering to dynamically retrieve only key information relevant to the query. Extensive experiments using the 2WikiMultiHopQA dataset (N=500, 10 iterations) demonstrate that the proposed method achieves a Fact Recovery rate of 72.00%, a 6-fold improvement over the truncation method. Furthermore, it validates high applicability in limited-context sLM environments through a 79% reduction in tokens and a 43% reduction in inference time. Comparative experiments between Gemma2 and Llama2 models also confirmed the method's versatility and reliability.**

***Index Terms*—Small Language Models, External Memory, Query-Aware Retrieval, Context Window Optimization, Multi-hop Reasoning**

## I. Introduction

Large Language Models (LLMs) demonstrate superior performance, but due to high computational costs and privacy concerns, lightweight Small Language Models (sLMs) are preferred in resource-constrained environments. Models such as Google's Gemma2 [2] and Meta's Llama2 [3] exhibit excellent performance with fewer than 7B parameters. However, they share a common limitation: a physically reduced context window due to hardware constraints. This acts as a critical weakness in tasks requiring long-term interaction with users.

Existing systems primarily adopt truncation policies that delete old memories or summarization policies that compress information. However, these approaches cause the permanent loss of key facts or semantic distortion, inducing hallucinations in the model. To resolve this, this study introduces the "Selective Attention" [1] mechanism, a human cognitive structure, into sLM memory management. Just as humans do not retain all information in working memory but retrieve it from long-term memory as needed, this pipeline stores extensive dialogue logs in external storage and dynamically activates only the information required for a query. Thus, while this study does not involve direct hardware-level experiments, it presents a practical solution for sLM environments by maximizing processing efficiency within a limited context window.

## II. Related Work

### A. sLMs and the Long Context Dilemma

Modern sLMs are parameter-efficient but have strict input limits, often around 2,048 tokens. While techniques like Sliding Window Attention [4] have been proposed, they require architectural changes, making universal application difficult. Therefore, a prompt engineering approach that optimizes input externally is more practical.

### B. Comparison with RAG

Retrieval-Augmented Generation (RAG) is a technology that retrieves external documents for LLMs [5], and there are attempts to apply this to dialogue logs. Recent surveys on Agentic RAG [13] highlight the importance of dynamic retrieval strategies that adapt to query context. While sophisticated techniques like Dense Retrieval exist, most focus on single-document-based retrieval, limiting their effectiveness in retrieving information based on dialogue flow. This study overcomes this limitation by introducing a window-based extraction method that considers the time-series characteristics of dialogue data.

### C. Comparison with Long Context Models

Recently, systems like RecurrentGPT [6], LongMem [7], LlamaIndex's ChatMemory [8], and Mem0 [12] have attempted to solve the long-term memory problem. Recent

surveys [14] categorize agent memory into factual, experiential, and working memory, emphasizing the importance of dynamic memory retrieval. However, these approaches are unsuitable for resource-constrained sLM environments as they require model architectural modifications or incur massive computational costs. In contrast, this study's pipeline adopts a lightweight post-processing method that operates without modifying the base model, enabling effective memory expansion at a low cost.

## III. Methodology: Query-Aware External Memory Pipeline

The proposed pipeline adopts a "Lazy Evaluation" approach, dynamically constructing context at the moment a user's query is input. The entire system consists of three stages: (1) Importance Extraction, (2) Embedding-based Filtering, and (3) Context Composition.

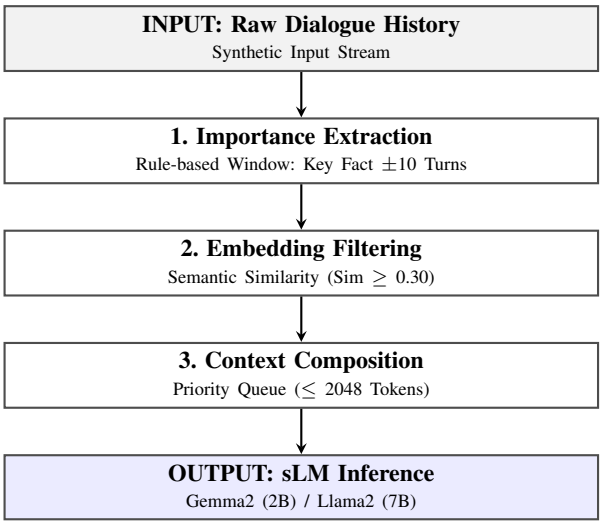


Fig. 1. Overall structure of the proposed Query-Aware External Memory Pipeline.

### A. Importance Extraction

Over 60% of dialogue consists of chit-chat with low informational value. To reduce LLM call costs, this study uses a rule-based approach to extract 10 turns before and after a section where a key fact appears ($Window = \pm 10$).

Specifically, Named-Entity Recognition (NER) is utilized to detect keywords and entities, identifying key facts. For example, if the utterance "Patient A stopped taking medication in February 2024" is input, the system detects entities like "Patient A (Person)", "February 2024 (Date)", and "medication (Concept)", selecting the utterance as an anchor for key information.

### B. Embedding-based Filtering

To remove noise irrelevant to the query from the initially selected information, semantic search is performed. Considering efficiency requirements, the lightweight `all-MiniLM-L6-v2` model [9] is used, and only information with a cosine similarity above the threshold $\tau$ is selected.

**Algorithm 1: Embedding Filtering Strategy**

**Input:** Query $Q$, Candidates $C = \{S_1, \ldots, S_n\}$
**Output:** Filtered Memory $M$

$v_Q \leftarrow \text{Encode}(Q)$ *// Query Vectorization*
$M \leftarrow \emptyset$
**for each** $S_i$ **in** $C$ **do**
  $v_S \leftarrow \text{Encode}(S_i)$ *// Sentence Vectorization*
  $sim \leftarrow \text{CosineSim}(v_Q, v_S)$
  **if** $sim \geq 0.30$ **then**
    $M \leftarrow M \cup \{S_i\}$
  **end if**
**end for**
**return** $M$

Through the above algorithm, only sentences with a cosine similarity score above the threshold are included in the final memory.

### C. Threshold Optimization Analysis

To find the optimal threshold, performance was measured while varying $\tau$ from 0.1 to 0.7.

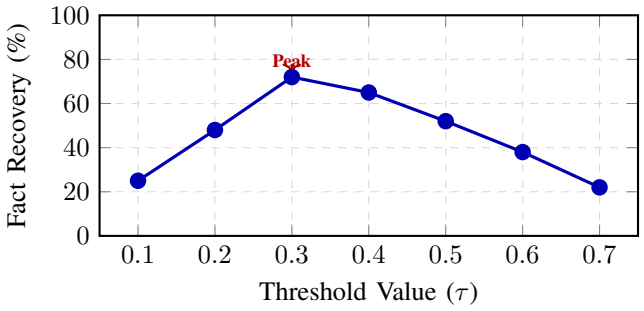


Fig. 2. Fact Recovery performance curve by threshold value. Performance peaks at $\tau = 0.30$, indicating optimal balance between noise removal and information preservation.

As shown in Fig. 2, performance peaks at $\tau = 0.30$, suggesting an optimal balance between noise removal and information loss prevention.

### D. Context Composition

Finally, the selected information is reassembled into the sLM's input prompt. To strictly adhere to the physical limit of 2,048 tokens, a priority queue strategy is used. An illustrative example of this composition is shown in Example 1.

Based on this hierarchical structure, the model is informed first of the objective (Query), followed by the immediate conversational context (Recent History), and finally reinforced with factual information (Retrieved Fact). This ensures optimal allocation of the limited token budget while preserving reasoning capability.

**Example 1. Priority-based Context Composition**

`[Priority 1]` Query: "When did A stop taking medication?"
`[Priority 2]` Recent History: "User: You mentioned not taking it yesterday." (Turn-2)
`[Priority 3]` Retrieved Fact: "Medication discontinuation date: 2024-02-15."

## IV. Experimental Design

### A. Dataset Selection and Sampling

This study selected 2WikiMultiHopQA [10] as the primary dataset. Compared to HotpotQA [11], it enforces Multi-hop Reasoning more strictly, making it suitable for verifying memory retrieval capabilities. For the main experiment, 500 samples were randomly extracted, and for the Ablation Study, a separate set of 100 samples was independently extracted. All processes were performed under a fixed random seed (Seed=42) for reproducibility.

### B. Experimental Environment and Statistical Verification

All experiments were conducted using a single NVIDIA RTX 3090 Ti (24GB VRAM) GPU and Ollama v0.3.12 environment. The main experiment was repeated 10 times under $N = 500$, $Seed = 42$ conditions to derive average values. An independent t-test was performed for statistical significance verification, confirming that the performance difference between Gemma2 and Llama2 was not statistically significant ($p \approx 0.087 > 0.05$).

### C. Evaluation Metrics

Fact Recovery is a binary metric indicating the proportion of key information essential for deriving the correct answer included in the prompt:

$$\text{Fact Recovery} = \frac{\text{Included Key Facts}}{\text{Total Key Facts}} \times 100 \tag{1}$$

Token Recall signifies the inclusion rate of correct answer tokens within the generated response:

$$\text{Token Recall} = \frac{|\text{Answer} \cap \text{Ground Truth}|}{|\text{Ground Truth}|} \times 100 \tag{2}$$

## V. Results and Analysis

### A. Main Performance Evaluation (N=500)

The results of the extensive experiment on the Gemma2 model are shown in Table I.

TABLE I
Main Experiment Results (Gemma2, N=500)

| Strategy | Recall | Recovery | Tokens | Latency |
|---|---|---|---|---|
| Truncation | 26.61% | 11.67% | 1,260 | 1.82s |
| Naive Sum | 25.15% | 9.17% | 396 | 3.05s |
| sLM-only | 58.45% | 42.67% | 316 | 0.62s |
| Proposed | 75.78% | 72.00% | 273 | 1.04s |

The proposed method achieved a Fact Recovery of 72.00%, showing a 6-fold performance improvement over Truncation (11.67%). Notably, Naive Summarization (9.17%) performed even worse than Truncation (11.67%), indicating that the summarization process itself causes semantic distortion and loss of critical factual details required for multi-hop reasoning. Additionally, the proposed method recorded 1.04 seconds in terms of Time Efficiency (Latency), demonstrating a favorable balance between performance and speed.

### B. Contribution Analysis (Ablation Study, N=100)

To analyze the contribution of each component of the proposed pipeline, removal experiments were conducted using separate samples (N=100). Results are shown in Table II.

TABLE II
Ablation Study Results (N=100)

| Configuration | Token Recall | Fact Recovery |
|---|---|---|
| Proposed (Full) | 68.32% | 61.17% |
| w/o Embedding | 57.84% | 43.67% |
| w/o Importance | 42.18% | 41.17% |
| Baseline (w/o Both) | 20.66% | 8.67% |

The numerical difference between the results in Table II and Table I (72.00% vs 61.17%) is attributed to sampling differences. The Ablation Study utilized 100 independent samples that did not overlap with the main experiment. Removing embeddings resulted in a performance drop of approximately 17.5 percentage points, and removing importance extraction caused a drop of about 20 percentage points, demonstrating the necessity of both modules. The combined effect of both modules (52.5 percentage points improvement from Baseline to Full) exceeds the sum of individual contributions, suggesting a synergistic interaction where importance extraction pre-filters noise, thereby enhancing the precision of subsequent embedding-based retrieval.

### C. Model Generalization Verification (Gemma2 vs Llama2)

To demonstrate that the pipeline is not overfitted to a specific model, the pipeline was applied to the Llama2 model under the same experimental conditions (N=500, 10 iterations). The Token Recall in Llama2 was 76.52%, showing a difference of only 0.74 percentage points from Gemma2 (75.78%), confirming that the difference is not statistically significant ($p > 0.05$). Thus, this pipeline can be considered a Model-Agnostic general technology independent of model architecture.

## VI. Conclusion

This study experimentally demonstrates that the memory deficiency problem of Small Language Models (sLMs) occurring in limited context environments can be effectively mitigated. The proposed Query-Aware External Memory Pipeline achieved up to a 6-fold improvement in information recovery rate compared to existing Truncation methods and confirmed

its practical value in small language model environments by simultaneously reducing token usage and inference time.

However, one limitation of this work is that the evaluation was performed on structured benchmark datasets with controlled experimental settings, and the hyperparameters (window size $\pm 10$, $\tau$=0.30) were not exhaustively tuned across different scenarios. Future research will address these gaps by conducting hyperparameter sensitivity analysis, investigating failure case patterns, and validating the pipeline's robustness on real-world noisy conversational data across diverse domains.

## ACKNOWLEDGMENTS

This work was also supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) – Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government (MSIT) (IITP-2025-RS-2024-00436773).

## REFERENCES

[1] A. M. Treisman, "Contextual cues in selective listening," Quarterly Journal of Experimental Psychology, vol. 12, no. 4, pp. 242–248, 1960.
[2] G. Team et al., "Gemma: Open models based on gemini research and technology," arXiv preprint arXiv:2403.08295, 2024.
[3] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv preprint arXiv:2307.09288, 2023.
[4] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," arXiv preprint arXiv:2004.05150, 2020.
[5] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, 2020.
[6] W. Zhou et al., "RecurrentGPT: Interactive generation of (arbitrarily) long text," arXiv preprint arXiv:2305.13304, 2023.
[7] W. Wang et al., "Augmenting language models with long-term memory," arXiv preprint arXiv:2306.07174, 2023.
[8] J. Liu, "LlamaIndex," 2023. [Online]. Available: https://github.com/run-llama/llama_index
[9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in Proc. EMNLP, 2019.
[10] X. Ho et al., "Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps," in Proc. COLING, 2020.
[11] Z. Yang et al., "HotpotQA: A dataset for diverse, explainable multi-hop question answering," in Proc. EMNLP, 2018.
[12] P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav, "Mem0: Building production-ready AI agents with scalable long-term memory," arXiv preprint arXiv:2504.19413, 2025.
[13] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei, "Agentic retrieval-augmented generation: A survey on agentic RAG," arXiv preprint arXiv:2501.09136, 2025.
[14] Y. Hu, S. Liu, et al., "Memory in the age of AI agents," arXiv preprint arXiv:2512.13564, 2025.