Bridging Intuitive Associations and Deliberate Recall: Empowering LLM Personal Assistant with Graph-Structured Long-term Memory

Anonymous ACL submission

Abstract

Large language models (LLMs)-based personal assistants may struggle to effectively utilize long-term conversational histories. Despite advances in long-term memory systems and dense retrieval methods, these assistants still fail to capture entity relationships and handle multiple intents effectively. To tackle above limitations, we propose **Associa**, a graph-structured memory framework that mimics human cognitive processes. Associa comprises an event-centric memory graph and two collaborative components: **Intuitive Association**, which extracts evidence-rich subgraphs through Prize-Collecting Steiner Tree optimization, and **Deliberating Recall**, which iteratively refines queries for comprehensive evidence collection. Experiments show that Associa significantly outperforms existing methods in retrieval metrics and user preference across dialogue benchmarks, advancing the development of more human-like AI memory systems.

1 Introduction

Empowered by large language models (LLMs), lifelong personal assistants have demonstrated remarkable potential across various domains, including daily life (Wu et al., 2025; Wang et al., 2024), healthcare (Jo et al., 2024; Zhang et al., 2024b), and mental health counseling (Zhong et al., 2024). These advancements show a significant opportunity to enhance quality-of-life and promote individual well-being through effective human-AI interaction (Li et al., 2024; Jo et al., 2024). However, the personal assistants face a critical challenge: effectively maintaining and utilizing lifelong conversational histories (Xu et al., 2022).

To tackle this challenge, researchers propose the long-term memory systems as a promising solution (Jo et al., 2024). These systems maintain the interaction history between assistants and users across multiple chat sessions (Xu et al., 2022), offering plug-and-play adaptability (Wu et al., 2025). Notably, studies have demonstrated that such memory-augmented systems can substantially outperform native long-context LLMs by effectively combining short-context reasoning accuracy with extended information retention capabilities (Maharana et al., 2024). Current implementations typically employ pre-trained dense retrieval approaches, enhanced with various forms of information such as summaries (Lu et al., 2023), facts (Wu et al., 2025), or observations (Maharana et al., 2024). However, these approaches exhibit significant deficiencies in their memory retrieval paradigms that hinder practical effectiveness (Yue et al., 2024; Wang et al., 2024).

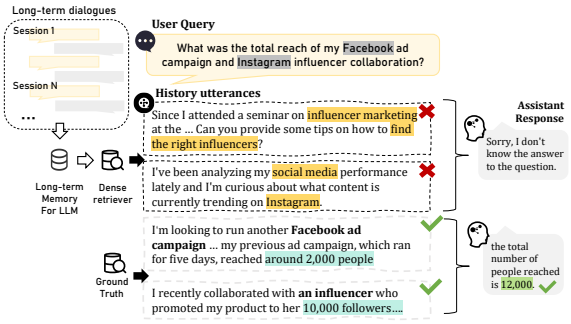

Figure 1: Illustration of Mismatching in Memory Retrieval for Long-term User Dialogues, especially the difference between dense retrieval approaches and the actual process of gathering evidence from users' long-term memory.

As shown in Figure 1, there is a mismatch between dense retrieval approaches and the actual process of gathering evidence from users' long-term memory (the memory retrieval approach). While dense retrievers are pre-trained to identify texts with the highest semantic similarity (Reimers and Gurevych, 2019), the goal of long-term memory systems is to effectively assist users in "gathering evidences" (Maharana et al., 2024). For instance,

when user poses the query: "What was the total reach of my Facebook ad campaign and Instagram influencer collaboration?," a pre-trained linear retrieval system might return historical user utterances related to "influencer marketing" or "social media." Although dense retrieval search documents show significant semantic similarity and may even include matching entities, they still do not provide the key evidence needed to answer the query. This reflects the two gaps between dense retrieval approaches and memory retrieval approaches: **(1) The neglect of the association of entities.** This instance denotes that the dense retriever ignores the binding relationship between "Facebook" and "ad campaign," as well as between "Instagram" and "influencer collaboration." In other words, utterances that include both of these entities will be considered stronger evidence. **(2) The dense retriever may mix and blur different evidence-gathering intents.** In this instance, the query contains two retrieval intents: "the reach of Facebook ad campaign" and "the reach of Instagram influencer collaboration." This overlap can cause the model to retrieve irrelevant information, undermining the accuracy of the results.

Inspired by the associative and deliberative nature of human long-term memory (Yue et al., 2024), we propose **Associa**, a novel graph-structured long-term memory framework, to enhance the ability to extract "key evidences" from very long-term dialogues, ultimately improving the effectiveness of the personal assistant. Specifically, to address gap (1), we first design an event-centric personal memory graph that incorporates multi-dimensional information. Then, we introduce an "Intuitive Association" retrieval module. This module employs Prize-Collecting Steiner Tree (PCST) optimization (He et al., 2025) with dynamic prize mechanisms, constructing memory subgraph by maximizing the prizes of subgraph nodes and minimizing the costs of subgraph edges. Through the subgraph, we retrieve utterances from long-term memory that are connected to the most "high-quality" nodes (such as events or entities). To address gap (2), we develop a "Deliberating Recall" module, which recursively assesses whether all evidences relevant to the user's intent have been fully collected. By instruction-tuning a specialized deliberating model, it collects missing clues and augment user's query as feedback. These two modules work collaboratively to ensure comprehensive evidence gathering.

Our contributions are threefold:

(1) We propose Associa, the first event-centric graph memory framework that systematically organizes long-term dialogue history by proposing a unified graph schema. The novel PCST-based subgraph retrieval mechanism enables associative memory retrieval in personal assistant systems, effectively addressing the information fragmentation challenge in extended conversations.

(2) Our innovative integration of Intuitive Association with Deliberating Recall establishes a human-like reasoning paradigm. The collaboration between two modules ensures the graph-structured memory will be selectively modified to gather more complete evidences.

(3) Extensive experiments results are conducted across several long-term personalized datasets, demonstrating that **Associa** achieves state-of-the-art performance in both recall accuracy and user preference metrics.

2 Related Work

2.1 Enhancing LLMs with long-term memory retrieval

In the context of lifelong personal assistants, user-assistant dialogues can accumulate extensive conversation histories over time. Given the limited context window of LLMs, processing the entire conversation history becomes impractical for long-term interactions (Jo et al., 2024). Existing research points out that commercial chat assistants and long-context LLMs show a 30% decline in accuracy of the benchmark when retaining information across ongoing interactions (Wu et al., 2025). There is also evidence that LLMs with long texts tend to hallucinate and recall information incorrectly (Maharana et al., 2024). A substantial body of evidence suggests that memory retrieval, compared to using base LLMs, can improve performance (Du et al., 2024; Kim et al., 2024). Its plug-and-play feature also makes it easy to integrate into other existing chat assistant systems (Wu et al., 2025). Accurate clue collection can significantly improve the performance of downstream tasks, such as question answering. However, there is still considerable potential for improvement in current retrieval methods (Kim et al., 2024).

2.2 Existing technical solutions of long-term memory retrieval

Current research mainly employs dense retrieval methods for memory retrieval. Specifically, these

methods retrieve the top-k relevant content from memory to enhance LLMs and design them for task adaptability. In addition to users' dialogue records, MemoryBank (Zhong et al., 2024) stores event summaries and dynamic personality understanding to help LLMs better understand users. LONGMEMEVAL (Wu et al., 2025) uses a series of techniques to improve memory retrieval, including fact concatenation, temporal filtering, and reasoning optimization (such as converting retrieval results into JSON format). Fragrel (Yue et al., 2024) splits texts into fragments, considering not only the similarity between the query and the fragments but also the similarity between the question and its context fragments.

However, these studies neglect the semantic relationships of user-related information. Our research proposes a graph-structured memory construction and associative retrieval approach to capture the relationships between memory chunks, thereby improving retrieval accuracy and efficiency.

3 Preliminary

3.1 Task definition

Long-term dialogue. Long-term dialogue L refers to the long-term interactions data between the user and the assistant (LLMs), often spanning across multiple sessions.

$$L = \bigcup_{i=1}^{n} S_i, \quad S_i = \langle r_1^{(i)}, r_2^{(i)}, ..., r_{k_i}^{(i)} \rangle \tag{1}$$

where S_i represents the i-th session, and k_i is the number of rounds in the i-th session.

Each session S_i contains an ordered sequence of dialogue rounds. For each single round $r_j^{(i)}$:

$$r_j^{(i)} = (u_j^{(i)}, a_j^{(i)}), \quad u_j^{(i)} \in \mathcal{U}, a_j^{(i)} \in \mathcal{A} \tag{2}$$

where $\mathcal{U}$ represents the user utterance space, and $\mathcal{A}$ represents the assistant utterance space.

Retrieval-augmented long-term memory generation. To respond a user's query q_t at timestamp t, the assistant must consider both the current session and relevant information from history dialogues. This retrieval-augmented long-term memory framework typically includes three core components (Zhang et al., 2024c): **memory management**, **memory retrieval**, **memory-enhanced response generation**. First, the **memory management** (Zhang et al., 2024c) constructs an external memory database $\mathcal{M}$ from historical long-term dialogues corpus L. This process includes memory writing, deletion, and editing, which can be formalized as:

$$\mathcal{M} = f_{\text{Manage}}(L) \tag{3}$$

where f_{Manage} represents the memory management function that transforms raw dialogue history into a memory database.

Subsequently, the **memory retrieval** module f_{retrieve} identifies and extracts relevant memory entries from $\mathcal{M}$ according to current q_t:

$$m_{q_t} = f_{\text{retrieve}}(\mathcal{M}, q_t) \tag{4}$$

where m_{q_t} represents the retrieved result to query q_t.

Finally, the LLM-based personal assistant **generates final response** r_t ny jointly considering the current session context S_t and above information.

$$r_t = f_{\text{LLM}}(q_t, S_t, m_{q_t}) \tag{5}$$

The advantage of external retrieval-based memory lies in two aspects: (1) higher interpretability. Users can clearly trace how the model retrieves information and makes decisions. This structure makes the source of knowledge and the reasoning process more transparent, enhancing the model's auditability and credibility. (2) External retrieval memory has higher transferability, which can be independent of specific model implementations.

4 The Proposed Framework: Associa

We propose **Associa**, a novel graph-structured long-term memory framework that enhances evidence extraction from extended dialogue histories and improves the effectiveness of personal assistants. Our framework introduces innovations in two crucial aspects: (1) an Event-centric personal memory graph for memory management, and (2) a collaborative memory retrieval mechanism that combines associative memory retrieval with deliberation recall.

4.1 Event-centric personal memory graph

To effectively manage the memory of user's daily events and interests, we design an event-centric personal memory graph that unifies graph schema and memory features. As shown in Figure 2, our approach is characterized by two key innovations: (1) **Event-centric memory architecture**: The graph organizes memories around event entities, leveraging their inherent rich contextual information (including temporal, spatial, and participant details)

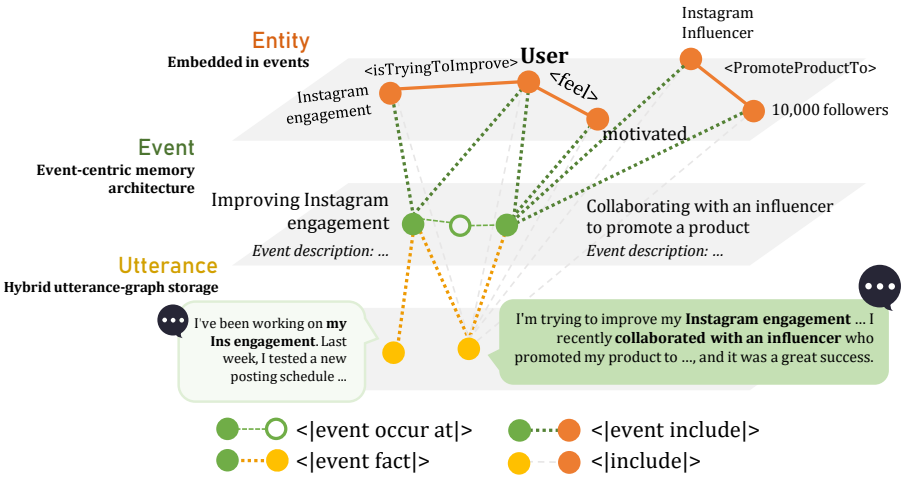

Figure 2: Illustration of the Event-centric Personal Memory Graph.

to create sophisticated memory structures. This design naturally captures the interconnected nature of personal experiences and facilitates complex relationship modeling through contextual anchors. (2) **Hybrid utterance-graph storage**: We establish explicit edges between graph nodes and their originating utterances in memory storage $\mathcal{M}$. This hybrid approach benefits both preserving critical raw information that might be lost during pure structured conversion, while simultaneously enably efficient graph-based retrieval operations.

Specifically, we propose a unified graph schema tailored for long-term memory in personal assistant scenarios. Based on common interaction patterns in personal assistance, our Associa memory graph formally designs two key components: **node types** and **egdes relationships**. **Node types**, The graph formally defines four core node types: utterance, user-related event, entity, and event time. The entity nodes represent referential objects embedded in events, encompassing diverse categories including [Object, Person/User/Organization, Resource, Place, Event, Goal/Intention, Time, Interest/Skill, Sentiment]. This comprehensive node type design extends beyond mere factual representation—it captures users' emotions, intentions, and preferences, thereby enabling personalized and empathetic assistance.

Egdes relationships: The graph topology is enriched through six semantically-typed edges that capture different aspects of user-assistant interactions:

(event, "<|event occur at|>", event time): The event time is inferred by the responding utterance timestamp.

(event, "<|event fact|>", utterance): It connects an event to an utterance that provides factual information about it, grounding events in user dialogue.

(event, "<|event include|>", entity): This edge associates an event with an entity involved in or affected by the event.

(utterance, "<|include|>", entity): The event contains entities, and the utterance is connected not only to the event but also to the entities within the event.

(user/speaker, "<|ask|>", utterance): The user is considered a key node because entities in the event are often related to "user," such as (user, browsing, yoga information), etc.

(entity, "<|relation|>", entity): The relationships are varied and can encompass actions, states, such as "occur at," or expressive verbs like "feel."

Graph deduplication: To enhance the efficiency and scalability of our memory graph, two deduplication strategy are used for events, entities, and relations. **Incremental deduplication:** based on FAISS, we take the advantage of the ability to dynamically and quickly merge duplicate nodes and edges as new memories are added (merging when the similarity exceeds the threshold). **Clustering deduplication:** we first build a similarity matrix and then applies Agglomerative Clustering for clustering. Its advantage is the ability to handle large-scale graphs in batches. The deduplication process helps reduce database storage space, and the merged events and entities contribute to associating more utterances, thereby optimizing graph-based retrieval.

4.2 Collaborative Memory Retrieval

To effectively retrieve relevant memories for user queries, we propose a Collaborative Memory Retrieval framework that synergistically combines associative intuition and deliberate recall, mimicking human memory retrieval processes. As illustrated in Figure 3, our framework operates in two complementary phases: **Intuitive Association:** Initially, we employ a subgraph retrieval mechanism that identifies potentially relevant information from historical dialogues memory graph, similar to human intuitive recall. **Deliberate Recall:** Recognizing that initial intuitive retrieval may overlook critical clues, we introduce a deliberate recall module that simulates human assistants' reflection process. This module systematically analyzes potential information gaps and augments the original query, enabling more comprehensive memory retrieval. Through iterative interaction between these two phases, our framework progressively refines the retrieved information.

4.2.1 Intuitive Association

We propose an enhanced approach for associative memory retrieval based on Prize-Collecting Steiner Tree (PCST) optimization with dynamic prize mechanisms, inspired by He et al. (2025). This novel method enables efficient extraction of relevant information from the user's historical dialogue memory graph. Our approach consists of three key phases:

Contextual Prize Initialization Given query embedding $q \in \mathbb{R}^d$ and memory graph $G = (V, E)$. With node feature $x^v \in \mathbb{R}^d$ and edge feature $e^{(u,v)} \in \mathbb{R}^d$. The cost of subgraph construction is a hyperparameter θ_{cost}. We compute initial relevance scores as **prizes** via:

$$Prize_n^v = \frac{q \cdot x^v}{\|q\|\|x^v\|}, Prize_e^{(u,v)} = \frac{q \cdot e^{(u,v)}}{\|q\|\|e^{(u,v)}\|}$$

Top-k selection with decaying weights:

$$\hat{Prize} = \begin{cases} k - \text{rank}(v) + 1 & \text{if } v \in \text{TopK}(R_n) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Add virtual nodes To overcome PCST's edge selection bias, we implement virtual node injection for high-prize edges:

$$\forall e_{uv} \in E : \begin{cases} \text{direct inclusion} & \hat{Prize}_e^{(u,v)} \leq \theta_{\text{cost}} \\ \text{insert virtual node } w & \hat{Prize}_e^{(u,v)} > \theta_{\text{cost}} \end{cases} \quad (7)$$

where virtual node w receives prize:

$$\hat{Prize}_n^w = \hat{Prize}_e^{(u,v)} - \theta_{\text{cost}} \quad (8)$$

Once the virtual nodes are added, node w constructs virtual edges with the two endpoints u and v, and the cost of the edge $C(e)$ is represented as follows:

$$C(e) = \begin{cases} 0 & \text{if } e \text{ is a virtual edge} \\ \theta_{\text{cost}} - \hat{Prize}_e^{(u,v)} & \text{otherwise} \end{cases} \quad (9)$$

Subgraph construction The objective of the PCST is to find a connected subgraph in a given graph such that the total prize of the selected nodes and edges minus the total cost is maximized:

$$\max_{\mathcal{T}} \left(\sum_{v \in \mathcal{T}} \hat{R}_n(v) + \sum_{e \in \mathcal{T}} \hat{R}_e(e) - \sum_{e \in \mathcal{T}} C(e) \right) \quad (10)$$

4.2.2 Deliberating Recall

We introduce the Deliberating Recall mechanism for the following reasons: (1) During the retrieval process, certain pieces of information in the query are crucial, but in a single round of retrieval, key details may be blurred or overlooked. A careful recall mechanism is needed to prompt the model to focus on these critical pieces of information. (2) Subgraph extraction allows for the retrieval of a connected subgraph. However, the user's intent in the query may be multifaceted and dispersed. Therefore, multiple rounds of recalling relevant clues are necessary to reconstruct the full context of the facts and accurately address the user's query.

Instruction Tuning for Deliberating model

To enhance the deliberation process, we fine-tune a specialized deliberating model that effectively identifies missing contextual cues from initial retrieval results, thereby improving the precision and efficiency of subsequent memory access.

Specifically, we leverage CoQA (Reddy et al., 2019), a well-established conversational question-answering dataset, to carefully construct training data for our Deliberating model. The data preparation process encompasses the following crucial steps: Graph Construction, Question Restoration, Locating the correct cues in the graph, Constructing positive and negative sample inputs and outputs (detailed description can be seen in Appendix A) [1] . We implement this training pipeline using

[1]The CoQA dataset is exclusively used for training the Deliberating Recall capability and remains completely independent from the datasets used for evaluating the overall system performance, ensuring unbiased assessment.

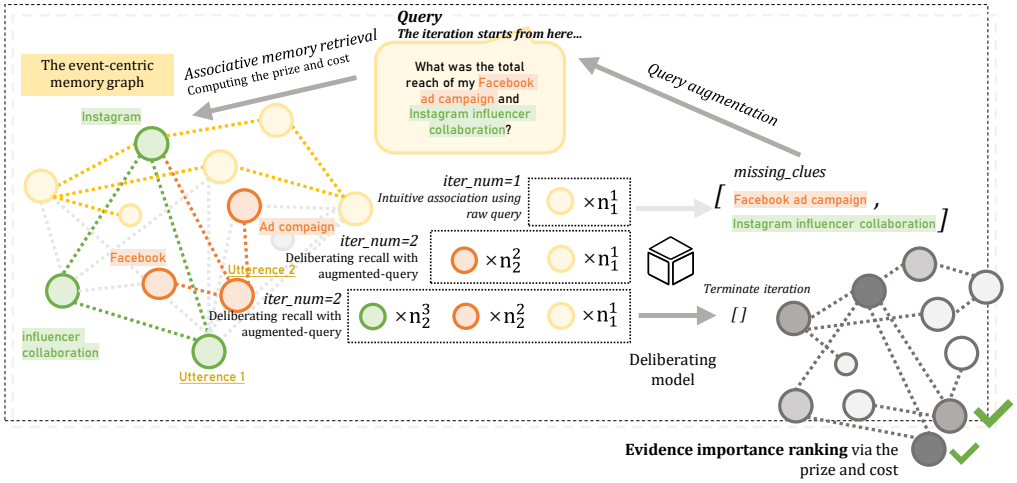

Figure 3: Framework of the Collaborative Memory Retrieval, consisting of two key modules: **Intuitive Association** and **Deliberate Recall**.

the Qwen-2.5-3B-instruct model as our base architecture. This approach significantly enhances the model's capability to identify and complement missing evidence.

Recursive evidence retrieval

To collect and complement the scattered clues, we use a recursive clue retrieval mechanism. Specifically, we optimize the focus of the retrieval question based on the current round of retrieval results and feedback from the deliberating model. This allow us to conduct a new round of subgraph retrieval, adding the nodes that were missed in the previous round. Through this systematic iteration, we achieve both comprehensive evidence consolidation and enhanced factual reconstruction fidelity. The algorithm is shown in Appendix B. Finally, we use the complete subgraph to calculate the evidence importance ranking (see Section 4.2.3).

4.2.3 Evidence importance ranking

To better identify key clues and assess node importance in our subgraph (consisting of utterances, user-related events, and entities), we propose using personalized PageRank to calculate customized weights for each node. Where $\mathbf{r}_i$ is the importance of nodes, $\mathcal{N}(i)$ is the neighbors of the node i. $w^{(i,j)}$ is the weight from node j to node i. d_j is the out-degree of node of j. p_i is the personalized preference of node i.

$$\mathbf{r}_i = \alpha \sum_{j \in \mathcal{N}(v)} \frac{w^{(i,j)}}{d_j} \mathbf{r}_j + (1-\alpha) p_i \tag{11}$$

In Associa, the p_i is the $\hat{prize}$ of nodes and the $w^{(i,j)}$ is formulated as follows:

$$w^{(i,j)} = \frac{1}{1 + \log\left(1 + C(e^{(i,j)})\right)} \tag{12}$$

5 Experiment

To gain more insights into Associa, we tend to address the following research questions (RQs) in this section.

RQ1: How does Associa perform in retrieval for long-term dialogue understanding?

RQ2: How does Associa perform in QA tasks for long-term dialogue understanding?

RQ3: What functions do the various modules of Associa serve in its performance?

5.1 Experiment setup

5.1.1 Datasets

To demonstrate the comprehensive capabilities of Associa, we test it on two datasets: Longmemeval$_s$ and Longmemeval$_m$.

Longmemeval Longmemeval (Wu et al., 2025) is a benchmark dataset designed to evaluate the very long-term memory capabilities of LLM-driven chat assistants. It contains 500 designed questions embedded within scalable user-assistant chat histories.

Longmemeval$_s$ with approximately 115k tokens per question (around 200 turns of dialogue) and **Longmemeval$_m$** with 1.5 million tokens per question (around 2000 turns of dialogue, 500 sessions). It tests the assistant's ability to perform five core

long-term memory tasks during sustained interactions: information extraction, multi-session reasoning, temporal reasoning, knowledge updating, and abstention.

5.1.2 Baselines

To validate the effectiveness of Associa, we compared it with several representative models in the retrieval task. The models used for comparison are as follows: **BM25**, a widely used text retrieval algorithm that evaluates document relevance to a query based on term frequency (TF) and inverse document frequency (IDF); **BGE-m3** (Chen et al., 2024), a retrieval model that achieves state-of-the-art performance in long-document retrieval; **Stella (stella_en_1.5B_v5)** (Zhang et al., 2024a), which utilizes a multi-stage distillation framework to reduce model size and vector dimensionality while maintaining high performance on text embedding benchmarks; **Contriever** (Izacard et al., 2021), which explores the potential of contrastive learning for training unsupervised dense retrievers and demonstrates strong performance on the BEIR benchmark; and **Longmemeval** (Wu et al., 2025), which enhances retrieval performance by combining Stella (1.5B) with user fact information and using LLMs for temporal filtering.

In the QA task, we compared **MemoRAG** (Izacard et al., 2021), an innovative RAG framework built on top of a highly efficient, super-long memory model. It utilizes a long-text memory model to provide an overview of the database, thereby optimizing retrieval results. **Longmemeval** (Wu et al., 2025) uses a retrieval-augmented approach, optimizing generation results in the generation phase through methods like CoN with JSON format. **Memorybank** (Zhong et al., 2024) integrates the Llama-index retriever, performing vectorized retrieval of documents and using large models to summarize the retrieved information, thereby enhancing the understanding of long-term memory for personal assistants.

5.2 Experiment metrics

For the retrieval task, this paper uses four metrics for evaluation: recall@5, recall@10, ndcg@5, and ndcg@10. The recall metric is defined as the retrieval of all memories, meaning that the recall for that sample is 1.

For the QA task, we use GPT-4o-mini for correctness evaluation. By inputting the task category, question, correct answer, and the model's generated response, GPT-4o-mini will return either correct or incorrect. Based on this, we evaluate the accuracy of the QA task.

5.2.1 Implementation details

Baseline implementation: In the baseline models for both retrieval generation and generation tasks, the methods from the original code repositories were adapted and implemented. Some retrieval-augmented methods require the use of retrievers. The choice of retriever was made based on the initial setup of the baseline models. For example, Longmemeval uses the Stella (1.5B) retriever, while MemoRAG uses the BGE-M3 retriever. Due to the long text in Longmemeval$_m$, the $beacon\_ratio$ in MemoRAG is set to 16, while for Longmemeval$_s$, it is set to 4. In the QA task, we use GPT-4o (gpt-4o-2024-11-20) as the generative model for testing.

In the execution of Associa, following the task setup of Wu et al. (2025), we only use "user-side" information and exclude data that cannot be recalled from the user information in a small number of cases. Associa uses Contriever as the dense retriever. Additionally, the following hyperparameter settings were chosen: for deliberating recall, max_iter is set to 2; in the intuitive association module, $cost_e$ is set to 0.5, and top_k (node setting) and top_e (edge setting) are set to 15. All experiments were conducted on a single Nvidia A800 (80GB) GPU.

5.3 R1: The performance of memory retrieval

In Table 1, the result has shown that (1) Associa with max_iter=2 achieves the highest overall performance across all metrics, excelling particularly in recall@5, recall@10, ndcg@5, and ndcg@10. The increased number of iterations in this version improves the model's ability to retrieve relevant results and rank them more effectively, making it the best performer in the experiment. On the other hand, Associa with max_iter=1 still shows a significant improvement, especially in recall@5 and ndcg@5, where it performs near the top of the table. Associa uses Contriever as the dense retriever. However, its performance far exceeds that of Contriever, further proving the effectiveness of this framework.

(2) The Longmemeval model, when combined with user facts and time filtering, demonstrates an improvement over BM25 and Contriever. While it does not reach the same level of performance

Models	Longmemeval-s				Longmemeval-m			
	recall@5	ndcg@5	recall@10	ndcg@10	recall@5	ndcg@5	recall@10	ndcg@10
BM25	0.507	0.538	0.591	0.567	0.383	0.424	0.456	0.451
Contriever	0.673	0.690	0.848	0.736	0.488	0.528	0.647	0.579
Stella	0.703	0.734	0.862	0.773	0.533	0.581	0.666	0.618
BGE	0.752	0.756	0.879	0.789	0.558	0.608	0.710	0.653
Longmemeval (w UF)	0.664	0.675	0.846	0.721	0.554	0.574	0.720	0.621
Longmemeval (w UF and TF)	0.701	0.720	0.867	0.759	0.563	0.592	0.722	0.634
Associa max iter=1	0.839	0.854	0.897	0.865	0.600	0.658	0.685	0.679
Associa max iter=2	**0.867**	**0.868**	**0.925**	**0.880**	**0.664**	**0.702**	**0.766**	**0.727**

Table 1: Performance for different models on two datasets. UF (User Fact) and TF (Time Filtering) are the specific features of baseline LongmemEval.

as Associa, BGE, or Stella, it still shows positive effects, particularly at recall@10 and ndcg@10. This suggests that incorporating user-related factors contributes positively to model performance.

(3) The difficulty of this task is fully reflected in longmemeval$_m$, as the scale of the dataset is enormous, making it akin to finding a needle in a haystack when identifying memory clues. Most models perform around 0.5 in the recall@5 metric. However, Associa effectively improves the recall of relevant evidence through its graph connection ability and recursive clue recall. Additionally, due to its evidence importance ranking capability, the model achieves optimal performance in terms of NDCG.

5.4 R2: The performance of question and answering

The result in Table 2 has shown that for understanding long-term memory, Associa demonstrates superior performance, highlighting the significant importance of enhanced retrieval in answering questions. Longmemeval, due to its integration of technologies such as CoN with JSON format, shows high effectiveness and performance in generation results. MemoryBank and MemoRAG perform poorly, possibly because excessively long text can reduce the comprehension ability of large language models when handling long texts.

Models	Longmemeval-s	Longmemeval-m
MemoRAG	0.05	0.06
Longmemeval (w UF)	0.80	0.64
Longmemeval (w UF and TF)	0.80	0.64
MemoryBank	0.26	0.12
Associa (iter=2)	**0.81**	**0.66**

Table 2: Performance on QA task.

Models	recall@5	ndcg@5	recall@10	ndcg@10
w/o AM	0.673	0.690	0.848	0.736
w/o DM	0.838	0.853	0.897	0.865
w/o SFT	0.841	0.860	0.913	0.873
w/o EIR	0.518	0.505	0.852	0.599
Associa	**0.867**	**0.868**	**0.925**	**0.880**

Table 3: Ablation test. w/o AM means w/o association mechanism, w/o DM means w/o deliberating module, w/o SFT means w/o fine-tuning of deliberating model, EIR means w/o evidence importance ranking.

5.5 R3: The ablation test for Associa

We conducted four ablation studies by removing key components of Associa: the association mechanism (AM), deliberating module (DM), specialized fine-tuning (SFT), and evidence importance ranking (EIR). As shown in Table 3, all variants showed performance degradation compared to the complete model, with the removal of the association mechanism causing the most significant drop. These results demonstrate that each component contributes to Associa's effectiveness, and their integration is crucial for optimal memory retrieval performance.

6 Conclusion

This work addresses the critical challenge of long-term conversational memory utilization in LLM-based personal assistants. We propose Associa, a cognitively inspired framework that overcomes the limitations of dense retrieval through two key innovations: (1) an event-centric graph memory preserving entity relationships, and (2) dual retrieval modules combining associative pattern matching with deliberate reasoning. Experimental validation across multiple benchmarks demonstrates Associa's superior performance. Our findings establish graph-structured memory with human-like retrieval mechanisms as a promising direction for developing AI capable of truly human-AI interaction.

Limitations

In our retrieval approach, we did not specifically model temporal information, which could be seen as an area for potential future enhancement. Additionally, our evaluation was limited to English-language datasets, and the assessment and learning of large models and agents would benefit from validation across a broader range of languages and corpora.

References

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai Wong. 2024. PerLTQA: A personal long-term memory dataset for memory classification, retrieval, and fusion in question answering. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 152–164, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2025. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Eunkyung Jo, Yuin Jeong, SoHyun Park, Daniel A Epstein, and Young-Ho Kim. 2024. Understanding the impact of long-term memory on self-disclosure with large language model-driven chatbots for public health intervention. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–21.

Jiho Kim, Woosog Chay, Hyeonji Hwang, Daeun Kyung, Hyunseung Chung, Eunbyeol Cho, Yohan Jo, and Edward Choi. 2024. Dialsim: A real-time simulator for evaluating long-term multi-party dialogue understanding of conversational agents. *arXiv preprint arXiv:2406.13144*.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, Rui Kong, Yile Wang, Hanfei Geng, Jian Luan, Xuefeng Jin, Zilong Ye, Guanjing Xiong, Fan Zhang, Xiang Li, Mengwei Xu, Zhijun Li, Peng Li, Yang Liu, Ya-Qin Zhang, and Yunxin Liu. 2024. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 13851–13870. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Zheng Wang, Zhongyang Li, Zeren Jiang, Dandan Tu, and Wei Shi. 2024. Crafting personalized agents through retrieval-augmented generation on editable memory graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4891–4906.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5180–5197, Dublin, Ireland. Association for Computational Linguistics.

Xihang Yue, Linchao Zhu, and Yi Yang. 2024. FragRel: Exploiting fragment-level relations in the external memory of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16348–16361, Bangkok, Thailand. Association for Computational Linguistics.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2024a. Jasper and stella: distillation of sota embedding models. *arXiv preprint arXiv:2412.19048*.

Kai Zhang, Yangyang Kang, Fubang Zhao, and Xiaozhong Liu. 2024b. Llm-based medical assistant personalization with short-and long-term memory coordination. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2386–2398.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024c. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

A Deliberating Recall Instruction Tuning Dataset Procession

CoQA (Reddy et al., 2019) containing 127,000 question-answer pairs from 8,000 dialogues. The dataset was processed as follows: (1) **Graph construction:** We extracted information from the 8,000 reading materials according to Section 4.1. For each dialogue, we created a graph that contains original text chunks, events, and entity-type nodes within events. (2) **Question restoration:** Since the dataset contains a lot of pronouns in the questions, we used a large model (qwen-plus) to restore the questions. For example, "Where is the location of this museum?" is restored, considering the reading material, to "Where is the location of The Vatican Apostolic Library?" (3) **Locating the correct cues in the graph:** CoQA provides cues based on the reading material. Using semantic similarity calculation methods, we locate the 2-3 graph nodes with the highest semantic similarity to the correct cues as T. (4) **Constructing positive and negative sample inputs:** We constructed two types of sample inputs. Positive samples include the correct cue as $Pos_{input}(q, \tilde{V})$, and negative samples exclude the correct cue as $Neg_{input}(q, V\tilde{\setminus}T)$ (where V represents the sampled graph nodes, and $V\tilde{\setminus}T$ represents the result of sampling the nodes from the graph after removing the correct cue T). (5) **Constructing positive and negative sample outputs:** For positive samples, we required the model to output an empty dictionary "{}". For negative samples, the model was asked to output the missing entities and indicate irrelevant content in the existing cues. We used qwen-plus to generate the samples. (6) **Model training:** We fine-tuned the qwen2.5-3B-instruct model for training.

B Deliberating Process Algorithm

Algorithm 1: Deliberating process

Definition : Retrieved synthetical graph $SynSubGraph$

Data: Max iter time max_iter, question q, Associa retriever $\mathcal{A}$, deliberating model $\mathcal{D}$, Memory Graph G, Evidence importance ranking EIR

def $SynSubGraph$, $iter\_time$: *int* $\rightarrow$ $SynSubGraph$**:**
  if $iter\_time == 0$ **then**
    $iter\_time \leftarrow iter\_time + 1$;
    $SynSubGraph \leftarrow \mathcal{A}(q, G)$;
  end
  if $iter\_time \geq max\_iter$ **then**
    return $SynSubGraph$
  end
  $missing\_clues \leftarrow \mathcal{D}(q, SynSubGraph)$;
  if *not* $missing\_clues$ **then**
    return $SynSubGraph$
  end
  for $missing\_clue \in missing\_clues$ **do**
    query $\leftarrow$ **query** $\oplus$ **missing_clue**;
    $SubGraph \leftarrow \mathcal{A}(query, G)$;
    $SynSubGraph \leftarrow expandSubgraph(SynSubGraph, SubGraph)$
  end
  return $(SynSubGraph, iter\_time + 1)$
$iter\_time \leftarrow 0$;
$SynSubGraph \leftarrow None$;
$SynSubGraph \leftarrow (SynSubGraph, iter\_time)$
$Nodes\_importance = EIR(SynSubGraph)$