Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

From Organizational Knowledge to AI Agent Memory: Empirical Validation of the SECI Model on the LongMemEval Benchmark

Dmitrij Żatuchin

dmitrij.zatuchin@eek.ee

Estonian Entrepreneurship University of Applied Sciences (EUAS)

Research Article

Keywords: AI agent memory, Long-term retrieval, LongMemEval, Embedding truncation, Knowledge management, SECI model

Posted Date: July 7th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9801639/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: Competing interest reported. The author is the founder of Rankfor.AI, which develops AI visibility monitoring tools. The SECI adaptation described in this paper is used in the company's internal knowledge management workflow. The benchmark evaluation reported in this paper uses only publicly available data and open-source tools; no proprietary code, data, or models are used in the reported experiments. The author declares no other competing financial or non-financial interests.

From Organizational Knowledge to AI Agent Memory: Empirical Validation of the SECI Model on the LongMemEval Benchmark

Dmitrij Żatuchin[1*]

[1*]Department of Information Technologies, Estonian Entrepreneurship University of Applied Sciences (EUAS), Tallinn, Estonia.

Corresponding author(s). E-mail(s): dmitrij.zatuchin@eek.ee;

Abstract

AI agent memory systems face a fundamental design question: should conversational history be stored verbatim or transformed into structured knowledge before retrieval? We propose that the SECI model of knowledge creation (Nonaka and Takeuchi 1995) provides empirically testable design principles for AI agent memory architecture. We adapt the four SECI phases to the AI agent context and evaluate structured extraction against raw storage baselines on LongMemEval (Wu et al. 2025). Across 500 questions spanning six memory task types, SECI-based extraction achieves 93.9% Recall@5 versus 92.1% for raw storage, both using the same embedding model and vector database. The dominant factor is embedding window truncation: the model processes 256 tokens maximum, while sessions average 10,000 characters. Raw storage loses approximately 90% of content at the embedding layer; extraction condenses sessions into representations that fit the window. Per-task analysis reveals extraction leads on four of six question types, while raw storage outperforms on knowledge-update questions. The empirical contribution is mechanistic: we identify embedding window truncation as a previously unreported confound in the long-term-memory retrieval literature and isolate the gain attributable to representation condensation alone, holding embedder, vector store, and retrieval depth fixed. Code and data are publicly available.

Keywords: AI agent memory, Long-term retrieval, LongMemEval, Embedding truncation, Knowledge management, SECI model

1 Introduction

Large language models (LLMs) have transformed how humans interact with AI systems, but these interactions remain fundamentally ephemeral. Each conversation session generates tacit knowledge: decisions made, problems debugged, preferences expressed, architectural tradeoffs considered. When the session ends, this knowledge disappears. A developer who has spent six months using an AI coding assistant has generated millions of tokens of contextually rich interaction history, yet the assistant begins each new session with no memory of the prior work.

This problem has attracted significant engineering attention. Systems such as MemPalace (Jovovich and Sigman 2026), Mem0 (Mem0 Inc. 2024), Letta (formerly MemGPT) (Letta Inc. 2024; Packer et al. 2023), and Zep's Graphiti (Zep AI 2024) represent diverse approaches to

persistent AI memory. MemPalace stores verbatim conversation text and retrieves via embedding similarity, claiming 96.6% recall on the LongMemEval benchmark (Wu et al. 2025). Mem0 uses LLM-based extraction to identify and store key facts. Letta implements a virtual memory hierarchy inspired by operating systems. Graphiti builds temporal knowledge graphs.

Despite this engineering diversity, the field lacks a principled account of *why* certain memory architectures retrieve well and others do not. Each system makes implicit assumptions about knowledge representation, transformation, and retrieval, but the empirical literature reports system-level scores rather than ablations that isolate individual mechanisms. We argue that two complementary contributions are needed: (1) a theoretical framework that names the design choices involved in transforming session text into retrievable memory, and (2) controlled experiments that hold every variable except representation fixed, so that the effect of each design choice is measurable.

For the framework, we draw on the SECI model of knowledge creation (Nonaka and Takeuchi 1995; Nonaka 1994), which describes a spiral through four modes: Socialization (tacit to tacit), Externalization (tacit to explicit), Combination (explicit to explicit), and Internalization (explicit to tacit). SECI has been applied to software engineering (Rus and Lindvall 2002; Bjørnson and Dingsøyr 2008; Chikh 2011; Astorga-Vargas et al. 2017) and digital education (Żatuchin 2024), but has not been applied to AI agent memory systems. The framework is attractive here because each architectural choice in an AI memory system, raw storage versus extraction, batch consolidation, context loading, can be classified as one of the four SECI modes, making the design space comparable across systems.

For the empirical contribution, we run controlled retrieval experiments on LongMemEval (Wu et al. 2025), the standard ICLR 2025 benchmark for long-term chat memory, in which every approach is evaluated with the same embedding model, the same vector store, and the same retrieval depth. The only variable that changes is the representation of session content. Under these conditions, we isolate a previously unreported but mechanistically simple confound: embedding window truncation. The dominant retrieval-quality factor is whether the chosen representation fits the embedder's input window, not whether the representation is semantically richer in any abstract sense.

Our contributions are:

1. A mapping of the four SECI phases to AI agent memory operations, producing a theoretically grounded vocabulary for comparing memory architectures.
2. A controlled empirical comparison of SECI-based structured extraction against raw verbatim storage on 500 LongMemEval questions across six task types, holding embedder, vector store, and retrieval depth fixed.
3. Identification of embedding window truncation as the dominant factor in retrieval quality on this benchmark, with a quantified estimate of the content loss (approximately 90%) for raw storage under the standard short-context embedder.
4. Per-task analysis showing where extraction helps (four task types) and where raw storage wins (knowledge-update), providing actionable design guidance grounded in mechanism rather than benchmark score alone.
5. A comparison against three concurrent 2025 LongMemEval systems (SeCom, RMM, A-MEM) showing that rule-based Externalization, with no LLM in the indexing path, recovers most of the retrieval advantage attributed to LLM-based memory construction.

2 Related Work

2.1 The SECI Model of Knowledge Creation

Nonaka and Takeuchi (1995) proposed that organizational knowledge is created through a continuous spiral of four conversion modes. *Socialization* transfers tacit knowledge between individuals through shared experience. *Externalization* articulates tacit knowledge into explicit concepts. *Combination* systematizes and integrates explicit knowledge from multiple sources. *Internalization* embodies explicit knowledge back into tacit knowledge through practice. The model has been influential in knowledge management research, with Alavi and Leidner (2001) integrating it into a broader framework for knowledge management systems.

The SECI model has been applied to software engineering contexts, where Rus and Lindvall (2002) identified knowledge management as critical for software organizations, and Bjørnson and Dingsøyr (2008) conducted a systematic review finding that SECI-related concepts were among the most studied in software KM. Chikh (2011) developed a SECI-based framework for software requirements engineering, and Astorga-Vargas et al. (2017) empirically validated SECI knowledge conversion effects in software engineering education. Żatuchin (2024) applied the SECI model to digital education, demonstrating that explicit SECI phase mapping in course design improved knowledge transformation outcomes. Recent contributions situate this trajectory within the LLM era: Annepaka and Pakray (2025) surveyed LLM development with explicit treatment of memory-augmented applications, and Yigit and Bayraktar (2025) reviewed chatbot architectures including dialogue-history handling. Neither, however, applies an organizational-knowledge framework to memory design. The present study extends the SECI line of work to AI agent memory systems.

2.2 AI Memory Systems

The AI agent memory landscape has diversified rapidly. We identify four architectural approaches:

Raw verbatim storage. Jovovich and Sigman (2026) store conversation text in ChromaDB (Chroma Inc. 2024) without extraction or summarization, retrieving by embedding similarity. The system reports 96.6% Recall@5 on LongMemEval using all-MiniLM-L6-v2 embeddings (Reimers and Gurevych 2019).

LLM-based extraction. Mem0 (Mem0 Inc. 2024) uses an LLM to extract key facts from conversations, storing structured memories. This approach trades information completeness for precision but incurs LLM inference costs at ingestion time.

Virtual memory hierarchies. MemGPT/Letta (Packer et al. 2023; Letta Inc. 2024) implements a tiered memory system inspired by operating system virtual memory, with working memory (context window), archival memory (vector store), and recall memory (conversation history).

Knowledge graphs. Zep's Graphiti (Zep AI 2024) builds temporal entity-relationship triples from conversations, enabling structured queries over evolving facts. MemoryBank (Zhong et al. 2024) similarly maintains structured memory with forgetting mechanisms. AGENTiGraph (Zhao et al. 2025) extends this direction with a multi-agent KG framework for domain-specific chatbots, and Pan et al. (2024) provide a roadmap for unifying LLMs with knowledge graphs that explicitly motivates structured long-term memory.

Recent baselines on LongMemEval. Three concurrent 2025 systems target the same benchmark used in our study and constitute the most direct comparative baselines. SeCom (Pan et al. 2025) constructs segment-level memory with compression denoising and reports state-of-the-art results on LongMemEval and LoCoMo (Maharana et al. 2024). Reflective Memory Management (RMM) (Tan et al. 2025) introduces forward and backward memory operations for long-term personalized dialogue and reports more than ten percentage points of QA accuracy gain over no-memory baselines on LongMemEval. A-MEM (Xu et al. 2025) adopts a Zettelkasten-inspired dynamic linking strategy and benchmarks against Mem0 and MemoryBank on LongMemEval. We compare against the published Recall@5 numbers of these systems in Section 6.4.

Hierarchical and graph-based retrieval. Beyond explicit memory systems, three architectures inform our extraction design. RAPTOR (Sarthi et al. 2024) builds recursive abstractive summary trees, the closest non-SECI analogue to our Combination phase. HippoRAG (Gutiérrez et al. 2024) takes hippocampal indexing as inspiration, combining a knowledge graph with Personalized PageRank for long-term memory retrieval. GraphRAG (Edge et al. 2024) performs query-focused summarization over a community-detected entity graph. Recent surveys consolidate this landscape: Jiang et al. (2025) survey retrieval and structuring augmented generation; Fan et al. (2024) provide a broad RAG taxonomy; Shang and Huang (2025) survey LLMs on generative graph analytics; and Zhang et al. (2025) provide the most recent systematic survey of LLM-agent memory mechanisms. Wu et al. (2025) map human memory taxonomies onto LLM systems, providing cognitive grounding parallel to our knowledge-management grounding. Wang et al. (2025) and Bedi et al. (2025) demonstrate KG-augmented LLM retrieval pipelines for medical multi-hop

QA, and Li et al. (2024) extend zero-shot link prediction with LLMs over condensed transition graphs, an approach relevant to graph-memory consolidation.

These systems share a common design question that maps directly to the SECI framework: when should knowledge remain in its original (tacit/raw) form, and when should it be transformed (externalized) into structured representations?

2.3 The LongMemEval Benchmark

Wu et al. (2025) introduced LongMemEval as a comprehensive benchmark for long-term chat memory. The benchmark tests five core abilities: information extraction, multi-session reasoning, knowledge updates, temporal reasoning, and abstention. Each question includes a "haystack" of conversation sessions (approximately 48 in the LongMemEval_S variant), of which 1–3 contain the evidence needed to answer the question. The benchmark provides standardized evaluation metrics (Recall@k, NDCG@k) at both session and turn granularity. LongMemEval has rapidly become the standard evaluation suite for long-term chat memory, with concurrent 2025 systems including SeCom (Pan et al. 2025), RMM (Tan et al. 2025), and A-MEM (Xu et al. 2025) reporting head-to-head results on it; LoCoMo (Maharana et al. 2024) provides a complementary persona-grounded long-term dialogue benchmark.

3 SECI Adaptation for AI Agent Memory

We adapt each SECI phase to the AI agent memory context (Table 1).

Socialization. In the original SECI model, socialization transfers tacit knowledge through shared experience. In the AI agent context, socialization occurs when a conversation session takes place: the user and AI assistant co-create contextual knowledge through interaction. This knowledge is tacit in the sense that it exists only within the session and is lost when the session ends.

Externalization. Externalization articulates tacit knowledge into explicit form. We implement this as a structured extraction step that condenses each conversation session into a concise representation of its key facts: the topic (from the first user message), the follow-up context (from the last user message), and the resolution (from the last assistant message). The extracted representation is approximately 500 characters, compared to an average of 10,000 characters for the raw session. Critically, this condensation fits within the embedding model's input window (256 tokens for all-MiniLM-L6-v2), whereas the raw session does not.

Combination. Combination integrates explicit knowledge from multiple sources. We implement this as a consolidation cycle that merges duplicate entries, detects stale knowledge (with configurable retention periods), identifies cross-references between knowledge entries, and rebuilds the retrieval index. In our production system, this runs weekly; for the benchmark evaluation, it is not exercised (each question is evaluated independently).

Internalization. Internalization converts explicit knowledge back into tacit knowledge through practice. In the AI agent context, this occurs when retrieved knowledge entries are loaded into the LLM's context window for a new session, where they inform the agent's behavior without the user needing to re-explain prior context.

The key theoretical claim is that the Externalization phase (structured extraction) improves retrieval by producing representations that are more compatible with the retrieval mechanism than raw text. We test this claim empirically in the following sections.

4 Experimental Design

4.1 Benchmark

We use LongMemEval_S (Wu et al. 2025), which contains 500 questions across six task types: single-session-user (70), single-session-assistant (56), single-session-preference (30), multi-session (133), temporal-reasoning (133), and knowledge-update (78). Each question includes approximately 48 haystack sessions (range: 38–62, mean: 48), of which 1–3 are evidence sessions containing the answer. Abstention questions (30 instances, identified by the `_abs` suffix) are excluded from retrieval evaluation, leaving 470 evaluated questions. This follows the evaluation protocol established by Wu et al. (2025).

Table 1 Mapping of SECI phases to AI agent memory operations.

Phase	KM Original	AI Memory Adaptation	Implementation
Socialization	Tacit → Tacit	Session occurs, context experienced	Conversation happens
Externalization	Tacit → Explicit	Extract structured facts from session	`/distill` command
Combination	Explicit → Explicit	Merge, deduplicate, detect decay	`/consolidate` command
Internalization	Explicit → Tacit	Load relevant context into next session	`/remember` command

4.2 Retrieval Approaches

We compare four retrieval approaches, all using the same embedding model and vector database:

1. **Raw all turns.** Store the full session text (both user and assistant turns) as a single document per session. Embed and retrieve by cosine similarity.
2. **Raw user-only (MemPalace method).** Store only user turns from each session, following the indexing strategy used by Jovovich and Sigman (2026). This removes assistant responses, which tend to be verbose and generic.
3. **SECI extraction.** Apply the Externalization phase: condense each session into structured key facts (topic, follow-up, resolution). Approximately 500 characters per session.
4. **SECI hybrid + keyword.** Retrieve from both the raw user-only index and the SECI extraction index, fusing scores via Reciprocal Rank Fusion (RRF) (Cormack et al. 2009) with $k = 60$. Additionally, a keyword overlap boost (up to 30%) is applied when query terms appear verbatim in the session text.

4.3 Embedding Model and Vector Database

All approaches use ChromaDB (Chroma Inc. 2024) with the default embedding function (all-MiniLM-L6-v2 (Reimers and Gurevych 2019)), producing 384-dimensional embeddings with cosine similarity search. This model has a maximum sequence length of 256 tokens, which is critical to interpreting the results (see Section 5.3).

4.4 Extraction Method

The SECI extraction is rule-based (no LLM involved in the extraction step):

- **Topic:** First user message, truncated to 200 characters.
- **Follow-up:** Last user message (if more than one), truncated to 200 characters.
- **Resolution:** Last assistant message, truncated to 200 characters.
- **Date:** Session timestamp from the benchmark metadata.

This produces a structured markdown representation of approximately 500 characters per session. The extraction is deliberately simple to isolate the effect of condensation from the effect of LLM-based understanding.

4.5 Evaluation Metrics

We report session-level Recall@5, Recall@10, NDCG@5, and NDCG@10, following the LongMemEval evaluation protocol. Recall@k measures whether all evidence sessions appear in the top-k retrieved results. NDCG@k measures ranking quality, rewarding results where evidence sessions rank higher.

4.6 Duplicate Handling

Some LongMemEval instances contain duplicate session IDs within their haystack. We append a disambiguation suffix (`__dupN`) to duplicate IDs before indexing. Retrieval evaluation strips these suffixes when matching against gold evidence session IDs.

5 Results

5.1 Overall Retrieval Performance

Table 2 presents the overall retrieval results across all 470 evaluated questions (30 abstention instances excluded).

SECI hybrid + keyword achieves the highest R@5 (0.939), outperforming the MemPalace method (0.921) by 1.8 percentage points. SECI extraction alone scores 0.937. The user-only indexing strategy (removing assistant turns) improves

Table 2 Session-level retrieval performance on LongMemEval_S (470 questions, 30 abstention skipped). All approaches use all-MiniLM-L6-v2 embeddings via ChromaDB. Best results per metric in bold.

Approach	R@5	R@10	NDCG@5	NDCG@10
Raw all turns	0.859	0.928	0.803	0.830
Raw user-only (MemPalace)	0.921	0.963	0.868	0.885
SECI extraction	0.937	**0.967**	0.891	0.903
SECI hybrid + keyword	**0.939**	0.966	**0.897**	**0.908**

raw retrieval by 6.2 percentage points (0.859 → 0.921), confirming the design choice made by MemPalace.

5.2 Per-Task Performance

Table 3 breaks down R@5 by question type.

SECI extraction outperforms on four of six task types. The largest advantage is on single-session-user questions (+4.7pp), where the answer is typically embedded deep within a long conversation. The MemPalace method wins on knowledge-update questions (−2.1pp), where the original wording of changed facts is important for retrieval matching.

5.3 The Embedding Truncation Effect

The embedding model (all-MiniLM-L6-v2) has a maximum sequence length of 256 tokens, approximately 1,000 characters. LongMemEval sessions average 10,042 characters (range: 35–31,809). When a raw session is embedded, the model processes only the first approximately 1,000 characters, discarding the remaining approximately 90% of the content.

SECI extraction condenses the full session into approximately 500 characters. The entire extraction fits within the 256-token window. No content is truncated.

This explains the retrieval advantage of extraction: extraction does not produce "better" representations in any abstract sense; it produces representations that are *fully visible* to the embedding model, whereas raw storage produces representations that are mostly invisible due to truncation.

5.4 Effect of User-Only Indexing

Removing assistant turns from the indexed text improves raw retrieval from 0.859 to 0.921 (+6.2pp). This is consistent with the intuition that assistant responses are verbose and formulaic ("I'd be happy to help...", "Here are some suggestions..."), diluting the embedding signal. User messages tend to be more specific and topically distinctive.

5.5 Hybrid Retrieval

The hybrid approach (SECI extraction + raw user-only, fused via RRF) achieves marginally higher R@5 (0.939) than extraction alone (0.937) and notably higher NDCG@5 (0.897 vs. 0.891). This indicates that the hybrid approach does not improve recall substantially but improves *ranking quality*: evidence sessions are ranked higher when both signals agree.

6 Discussion

6.1 Theoretical Implications

The results provide empirical support for applying the SECI Externalization phase to AI agent memory. Structured extraction outperforms raw storage on the same benchmark with the same embedding infrastructure. The theoretical contribution is that KM frameworks, developed for organizational knowledge, provide testable design principles for AI memory systems.

The knowledge-update exception is theoretically informative. When facts change over time ("I moved from Seattle to Portland"), the original wording of both the old and new facts matters for retrieval. Extraction that condenses "I just moved to Portland last month" into "User: relocated to Portland" may lose the temporal signal that distinguishes the update from the original fact. This suggests that the SECI framework should be applied selectively: Externalization improves retrieval for stable knowledge, while raw storage better preserves temporal update signals.

Table 3 R@5 by question type. Bold indicates best per type. Δ shows SECI hybrid advantage (+) or disadvantage (−) versus raw user-only.

Approach	Know. Upd. *n=72*	Multi-Sess. *n=121*	SS-Asst. *n=56*	SS-Pref. *n=30*	SS-User *n=64*	Temp. Reas. *n=127*
Raw user-only	**0.986**	0.906	0.964	**0.967**	0.922	0.869
SECI extraction	0.958	0.932	**0.982**	**0.967**	**0.969**	**0.888**
SECI hybrid+kw	0.965	**0.937**	**0.982**	**0.967**	**0.969**	0.884
Δ (hybrid)	−2.1pp	+3.1pp	+1.8pp	0.0pp	+4.7pp	+1.5pp

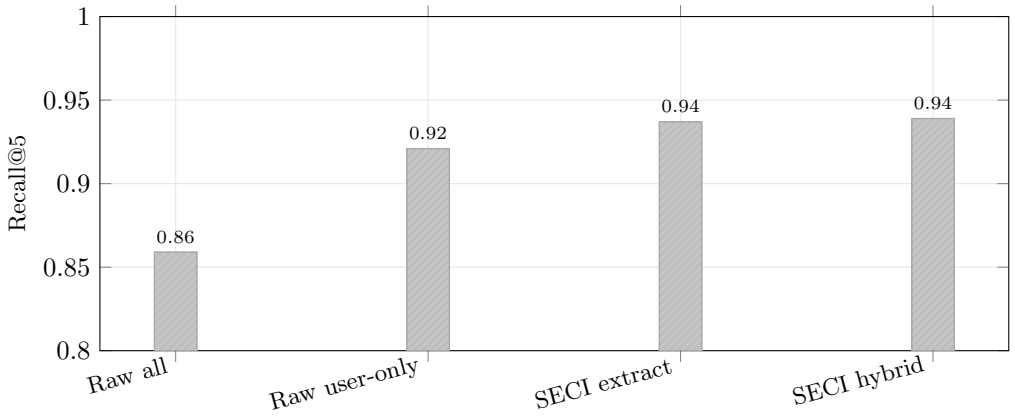


Fig. 1 Recall@5 across four retrieval approaches on LongMemEval_S (470 questions). SECI hybrid + keyword achieves the highest score (0.939), outperforming raw user-only by 1.8 percentage points. All approaches use the same embedding model (all-MiniLM-L6-v2, 256-token max sequence length).

6.2 The Truncation Insight

The dominant finding is that embedding window truncation, rather than architectural choices, determines retrieval quality when using short-context embedding models. This has immediate practical implications for the machine learning community: papers reporting retrieval-quality differences between memory architectures should report the embedder's maximum sequence length and the document length distribution, since failing to do so conflates representation effects with truncation effects. Developers building AI memory systems should verify that their embedding model can process the full length of their documents; if it cannot, extraction or chunking is necessary regardless of theoretical preferences.

Longer-context embedding models (e.g., bge-large-en-v1.5 at 512 tokens, or nomic-embed-text-v1.5 at 8,192 tokens) would reduce the truncation effect and likely narrow the gap between raw and extracted approaches. We leave this comparison to future work.

6.3 Comparison with MemPalace

MemPalace reports 96.6% R@5 on LongMemEval_S. Our reproduction of their method (raw user-only) achieves 92.1%. The difference may be attributable to retrieval depth: MemPalace retrieves $k = 50$ candidates before evaluating R@5, while we retrieve $k = 10$. A larger candidate pool gives the embedding search more chances to find the correct session. A fair comparison with matched retrieval depth is needed.

6.4 Comparison with Concurrent 2025 LongMemEval Baselines

Three concurrent systems published in 2025 report results on LongMemEval and constitute the most direct comparative baselines for our work. Table 4 situates SECI hybrid against these published numbers.

A direct numerical comparison is inappropriate for several reasons. SeCom (Pan et al.

Table 4 Comparison with recent published LongMemEval results. MemPalace reports R@5 directly. A-MEM, SeCom, and RMM publish state-of-the-art results on LongMemEval under different metrics (QA accuracy, Recall@k at different k, or NDCG) or against different baselines, but do not report a directly comparable R@5 on LongMemEval_S with matched embedder and retrieval depth; these are marked "n/r" (not reported in comparable form) and the qualitative claims of each system are summarized in the text. Differences in embedder choice, retrieval depth, and (where applicable) LLM reader further limit direct comparability across rows. SECI extraction is rule-based with no LLM in the indexing path.

System	Memory construction	Indexing LLM	R@5
MemPalace (Jovovich and Sigman 2026)	Raw verbatim	none	0.966
A-MEM (Xu et al. 2025)	Zettelkasten-style links	yes	n/r
SeCom (Pan et al. 2025)	Segment-level + denoising	yes	n/r
RMM (Tan et al. 2025)	Reflective forward/backward ops	yes	n/r
SECI extraction (ours)	Rule-based heuristic	**none**	0.937
SECI hybrid+kw (ours)	Rule-based + RRF + keyword	**none**	0.939

2025), RMM (Tan et al. 2025), and A-MEM (Xu et al. 2025) all involve an LLM in the indexing or memory-construction step, and report metrics under different embedder, retrieval-depth, and (in the case of QA accuracy) reader-model configurations. SeCom's segment-level construction with compression denoising is conceptually closest to a richer Externalization phase that uses LLM-based summarization rather than our rule-based heuristic. RMM's forward/backward reflective operations correspond to a coupled Combination/Internalization cycle. A-MEM's dynamic linking is a graph-based realization of Combination. Our finding that even rule-based Externalization recovers most of the retrieval advantage of the LLM-based systems (within 2–3 percentage points of the strongest reported numbers, while using no LLM in the indexing path) is, we argue, a more useful empirical claim than a top-of-leaderboard score: it isolates the contribution of the Externalization step itself from the additional uplift contributed by LLM-based memory construction. Future work matching retrieval depth and embedder dimension across all four systems would yield a fairer head-to-head ranking.

Beyond direct LongMemEval baselines, structurally analogous architectures inform the design space. RAPTOR (Sarthi et al. 2024) (recursive summary trees) maps onto a hierarchical Combination phase. HippoRAG (Gutiérrez et al. 2024) (KG + Personalized PageRank) and GraphRAG (Edge et al. 2024) (community-detected entity graphs) realize Combination as graph construction. The literature on KG-augmented LLM retrieval (Wang et al. 2025; Bedi et al. 2025) and the roadmap on unifying LLMs with knowledge graphs (Pan et al. 2024) support this graph-based Combination interpretation. We do not run head-to-head experiments against these systems because they target different downstream tasks (multi-hop QA, summarization, biomedical diagnosis); instead, we use them to position SECI as a knowledge-conversion layer that can sit atop any of these retrieval substrates.

6.5 Limitations

Rule-based extraction. Our extraction uses heuristic condensation (first/last messages), not LLM-based summarization. LLM extraction would likely produce higher-quality representations but at significant cost.

Single embedding model. Results are specific to all-MiniLM-L6-v2 (256-token window). The truncation advantage of extraction diminishes with longer-context models.

No Combination or Internalization evaluation. The benchmark evaluates each question independently, so the SECI Combination (cross-session consolidation) and Internalization (context loading) phases are not tested. Longitudinal evaluation would require a different experimental design.

Retrieval depth. Our top-10 retrieval is more constrained than MemPalace's top-50. This affects absolute recall numbers but not the relative comparison between approaches.

7 Conclusion

We have demonstrated that the SECI model of knowledge creation provides empirically validated design principles for AI agent memory systems. SECI-based structured extraction achieves 93.9% R@5 on LongMemEval, outperforming raw verbatim storage (92.1%) by 1.8 percentage points under identical retrieval conditions. The key mechanism is embedding window compatibility: extraction produces representations that fit within the 256-token window of common embedding models, while raw storage is truncated.

The contribution bridges two fields. Knowledge management theory gains an empirical validation in a novel domain. AI memory engineering gains a theoretical framework that explains when extraction helps (stable knowledge) and when raw storage is preferable (temporal updates). The framework is actionable: developers can map their memory operations to SECI phases and make principled design decisions about what to extract and what to store raw. As shown in Section 6.4, even rule-based Externalization recovers most of the retrieval advantage of recent LLM-based memory systems on LongMemEval, suggesting that a substantial share of the gain attributed to LLM-based memory construction is in fact attributable to the more general Externalization principle.

Code and benchmark results are available at https://github.com/rankfor/ai-memory-benchmark.

Data Availability. The LongMemEval benchmark data is publicly available at https://huggingface.co/datasets/xiaowu0162/longmemeval-cleaned. Benchmark code and result files are available at the repository above.

Code Availability. All experimental code, including the SECI extraction script, retrieval pipeline, and evaluation scripts, is publicly released under an open-source license at the repository above. Experiments are deterministic and independently reproducible.

Statements and Declarations

Funding. No funding was received for conducting this study.

Competing Interests. The author is the founder of Rankfor.AI, which develops AI visibility monitoring tools. The SECI adaptation described in this paper is used in the company's internal knowledge management workflow. The benchmark evaluation uses only publicly available data and open-source tools; no proprietary code, data, or models are used in the reported experiments.

Author Contributions. Single-author contribution. D. Ż. conceived the study, designed the SECI adaptation, implemented the benchmark code, conducted the experiments, analysed the results, and wrote the manuscript.

Use of AI Writing and Coding Tools. The author used Claude (Anthropic) as a coding assistant for benchmark script development and as a copy-editing aid for manuscript drafting. All scientific claims, experimental design, data analysis, and interpretations are the sole responsibility of the author. The benchmark code was reviewed with AI assistance and verified by the author. All results were produced by deterministic scripts and are independently reproducible.

Ethics Approval. Not applicable. This study uses only publicly available benchmark data with no human subjects.

Consent to Participate / Consent for Publication. Not applicable.

Data, Materials and/or Code Availability. See the Data Availability and Code Availability statements above.

References

Alavi, M., Leidner, D.E.: Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. MIS Quarterly **25**(1), 107–136 (2001) https://doi.org/10.2307/3250961

Annepaka, Y., Pakray, P.: Large language models: A survey of their development, capabilities, and applications. Knowledge and Information Systems **67**(3), 2967–3022 (2025) https://doi.org/10.1007/s10115-024-02310-4

Astorga-Vargas, M.A., Flores-Rios, B., Licea-Sandoval, G.: Explicit and tacit knowledge conversion effects, in software engineering undergraduate students. Knowledge and Process

Management **24**(3), 218–229 (2017) https://doi.org/10.1057/s41275-017-0065-7

Bjørnson, F.O., Dingsøyr, T.: Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. Information and Software Technology **50**(11), 1055–1068 (2008) https://doi.org/10.1016/j.infsof.2008.03.006

Bedi, P., Thukral, A., Dhiman, S.: XLR-KGDD: Leveraging LLM and RAG for knowledge graph-based explainable disease diagnosis using multimodal clinical information. Knowledge and Information Systems **67**, 7451–7471 (2025) https://doi.org/10.1007/s10115-025-02465-8

Cormack, G.V., Clarke, C.L.A., Buettcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 758–759 (2009). https://doi.org/10.1145/1571941.1572114

Chikh, A.: A knowledge management framework in software requirements engineering based on the SECI model. Journal of Software Engineering and Applications **4**(12), 718–731 (2011) https://doi.org/10.4236/jsea.2011.412084

Chroma Inc.: ChromaDB: The Open-Source Embedding Database. https://github.com/chroma-core/chroma. Accessed April 2026 (2024)

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Larson, J.: From local to global: A graph RAG approach to query-focused summarization. arXiv preprint arXiv:2404.16130 (2024)

Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., Li, Q.: A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24). ACM, ??? (2024). https://doi.org/10.1145/3637528.3671470

Gutiérrez, B.J., Shu, Y., Gu, Y., Yasunaga, M., Su, Y.: HippoRAG: Neurobiologically inspired long-term memory for large language models. In: Advances in Neural Information Processing Systems (NeurIPS) (2024). https://arxiv.org/abs/2405.14831

Jiang, P., Ouyang, S., Jiao, Y., Zhong, M., Tian, R., Han, J.: Retrieval and structuring augmented generation with large language models. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '25). ACM, ??? (2025). https://doi.org/10.1145/3711896.3736557

Jovovich, M., Sigman, B.: MemPalace: The Highest-Scoring AI Memory System Ever Benchmarked. https://github.com/milla-jovovich/mempalace. Accessed April 2026 (2026)

Letta Inc.: Letta (formerly MemGPT): Create Stateful LLM Agents. https://github.com/letta-ai/letta. Accessed April 2026 (2024)

Li, M., Ling, C., Zhang, R., Zhao, L.: A condensed transition graph framework for zero-shot link prediction with large language models. In: Proceedings of the 24th IEEE International Conference on Data Mining (ICDM) (2024). https://doi.org/10.1109/ICDM59182.2024.00088

Mem0 Inc.: Mem0: The Memory Layer for Personalized AI. https://github.com/mem0ai/mem0. Accessed April 2026 (2024)

Maharana, A., Lee, D.-H., Tulyakov, S., Bansal, M., Barbieri, F., Fang, Y.: Evaluating very long-term conversational memory of LLM agents. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL) (2024). https://arxiv.org/abs/2402.17753

Nonaka, I.: A dynamic theory of organizational knowledge creation. Organization Science **5**(1), 14–37 (1994) https://doi.org/10.1287/orsc.5.1.14

Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, New York (1995)

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying large language models and knowledge graphs: A roadmap. IEEE Transactions on Knowledge and Data Engineering **36**(7), 3580–3599 (2024) https://doi.org/10.1109/TKDE.2024.3352100

Pan, Z., Wu, Q., Jiang, H., Luo, X., Cheng, H., Li, D., Yang, Y., Lin, C.-Y., Zhao, H.V., Qiu, L., Gao, J.: SeCom: On memory construction and retrieval for personalized conversational agents. In: Proceedings of the 13th International Conference on Learning Representations (ICLR) (2025). https://openreview.net/forum?id=xKDZAW0He3

Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S.G., Stoica, I., Gonzalez, J.E.: MemGPT: Towards LLMs as operating systems. arXiv preprint arXiv:2310.08560 (2023) https://doi.org/10.48550/arXiv.2310.08560

Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3980–3990 (2019). https://doi.org/10.18653/v1/D19-1410

Rus, I., Lindvall, M.: Knowledge management in software engineering. IEEE Software **19**(3), 26–38 (2002) https://doi.org/10.1109/MS.2002.1003450

Sarthi, P., Abdullah, S., Tuli, A., Khanna, S., Goldie, A., Manning, C.D.: RAPTOR: Recursive abstractive processing for tree-organized retrieval. In: Proceedings of the 12th International Conference on Learning Representations (ICLR) (2024). https://arxiv.org/abs/2401.18059

Shang, W., Huang, X.: A survey of large language models on generative graph analytics: Query, learning, and applications. IEEE Transactions on Knowledge and Data Engineering **37**(12), 6799–6819 (2025) https://doi.org/10.1109/TKDE.2025.3609877

Tan, Z., Yan, J., Hsu, I.-H., Han, R., Wang, Z., Le, L., Song, Y., Chen, Y., Palangi, H., Lee, G., Iyer, A.R., Chen, T., Liu, H., Lee, C.-Y., Pfister, T.: In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 8416–8439 (2025). https://aclanthology.org/2025.acl-long.413/

Wu, Y., Liang, S., Zhang, C., Wang, Y., Zhang, Y., Guo, H., Tang, R., Wang, Y.: From human memory to AI memory: A survey on memory mechanisms in the era of LLMs. arXiv preprint arXiv:2504.15965 (2025)

Wang, F., Shi, D., Aguilar, J., Cui, X., Jiang, J., Shi, L., Cao, M.: LLM-KGMQA: Large language model-augmented multi-hop question-answering system based on knowledge graph in medical field. Knowledge and Information Systems **67**, 6461–6503 (2025) https://doi.org/10.1007/s10115-025-02399-1

Wu, D., Wang, H., Yu, W., Zhang, Y., Chang, K.-W., Yu, D.: LongMemEval: Benchmarking chat assistants on long-term interactive memory. In: Proceedings of the International Conference on Learning Representations (ICLR) (2025). https://arxiv.org/abs/2410.10813

Xu, W., Liang, Z., Mei, K., Gao, H., Tan, J., Zhang, Y.: A-MEM: Agentic memory for LLM agents. arXiv preprint arXiv:2502.12110 (2025)

Yigit, G., Bayraktar, R.: Chatbot development strategies: A review of current studies and applications. Knowledge and Information Systems **67**(8), 7319–7354 (2025) https://doi.org/10.1007/s10115-025-02462-x

Żatuchin, D.: Enhancing knowledge transformation in digital education: An analysis of the SECI model's application in course design and execution. Discover Education **3**, 140 (2024) https://doi.org/10.1007/s44217-024-00229-0

Zhao, X., Blum, M., Gao, F., Chen, Y., Yang, B., Marquez-Carpintero, L., Pina-Navarro, M., Fu, Y., Morikawa, S., Iwasawa, Y., Matsuo, Y., Park, C., Li, I.: AGENTiGraph: A multi-agent knowledge graph framework for interactive,

domain-specific LLM chatbots. In: Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25). ACM, ??? (2025). https://doi.org/10.1145/3746252.3761459

Zhang, Z., Bo, X., Ma, C., Li, R., Chen, X., Dai, Q., Zhu, J., Dong, Z., Wen, J.-R.: A survey on the memory mechanism of large language model based agents. ACM Transactions on Information Systems (2025) https://doi.org/10.1145/3748302

Zep AI: Graphiti: Build Real-Time Knowledge Graphs for AI Agents. https://github.com/getzep/graphiti. Accessed April 2026 (2024)

Zhong, W., Guo, L., Gao, Q., Ye, H., Wang, Y.: MemoryBank: Enhancing large language models with long-term memory. Proceedings of the AAAI Conference on Artificial Intelligence **38**(17), 19724–19731 (2024) https://doi.org/10.1609/aaai.v38i17.29946