

SOMA - Intelligence Transcends Memory, a Framework-Prioritized Cognitive Architecture for AI Agents

Author: Sun Yan

Abstract

Current research on memory in large language models and AI Agents primarily focuses on enhancing storage capacity and retrieval accuracy—ultimately aiming to emulate high-intelligence behavior. However, the truly insightful "high-intelligence" humans do not rely on photographic memory, but possess a foundational cognitive framework capable of analyzing all phenomena. They transform experiences and knowledge into "current knowledge reserves," retrieving them precisely through non-linear, hierarchical processes to address core issues. Inspired by the self-evolving mechanism of the open-source project EvoMap and M-Flow's associative graph retrieval, and integrating cognitive science principles of multi-memory systems with human cognitive development, this paper proposes SOMA (Somatic Wisdom Architecture): an integrated intelligent management system for retrieval and memory. Guided by a self-evolving cognitive framework engine, SOMA employs bidirectionally activated associative computation to interweave episodic and semantic memories into a dynamic "knowledge network," enabling the solidification of experience into skills and autonomous framework growth through metacognitive reflection. The system enables AI agents to not only "know extensively" but also "understand profoundly," achieving deep domain analysis and decision-making with minimal computational resources.

Keywords: Memory system, Thinking framework, Self-evolution, Associative retrieval, Agent, Intelligent management

1. Introduction

As large language models (LLMs) are increasingly applied in complex tasks, how to endow them with long-cycle, evolutionary memory has become a critical bottleneck on the way to general artificial intelligence. The existing technical approaches are roughly divided into two categories: first, leveraging vector databases for rapid retrieval of massive information to simulate an "immediate recall" memory; second, designing sophisticated memory graph architectures to enhance retrieval relevance and logic, as demonstrated by the M-Flow project's inverted conical directed graph associative retrieval. In addition, the EvoMap project encapsulates successful task paths as heritable, mutable "genes," enabling the self-evolution of Agent skills. These advancements have respectively made significant progress in the organization and evolution of memory.

However, the most valuable trait of human cognition—wisdom—is not entirely correlated with pure intelligence (IQ) or memory alone. A wise man may not have an exceptional memory, but he has a highly abstract, transferable underlying framework of thinking. When confronted with any problem, he first uses this framework to decompose the problem into several essential elements and boundaries, then rapidly and non-linearly draw upon accumulated life experiences and knowledge to address the current challenge, ultimately generating insightful solutions that cut to the core. This cognitive paradigm of "utilizing past resources to navigate the present" constitutes a memory management system indexed by mental frameworks and governed by associative potential. It completely eliminates passive data storage, enabling proactive and intelligent memory manipulation.

This paper aims to deeply integrate this human cognitive model with cutting-edge AI memory technologies to develop an integrated retrieval-and-memory management system—SOMA. SOMA does not seek to store every detail but instead

strives to create a "thinking operating system" comparable to human intelligence: it combines EvoMap's self-evolution capabilities with M-Flow's associative architecture under a higher-dimensional cognitive framework engine, unifying memory storage, forgetting, retrieval, and evolution to address immediate challenges. The following sections will first review relevant research insights, then refine the wise-person thinking model, present SOMA's complete architecture and core algorithms, and explore its applications in digital avatar and think tank management scenarios.

2. Related Research and Insights

2.1 EvoMap: Self-Evolving Memory and Skill Genes

The EvoMap project introduces a "genetic encapsulation" concept, abstracting agents' successful task trajectories into genes and achieving generational skill evolution through a "scan– mutate– verify– solidify" cycle. It maintains a three-tiered system of working memory, short-term memory, and long-term memory, with information of varying importance undergoing differentiated attenuation and consolidation processes. The core insight from EvoMap is that memory is not merely storage, but rather an evolving system capable of growth and elimination. Notably, transforming validated behavioral patterns into customizable "skill modules" directly simulates the formation of procedural memory.

2.2 M-Flow: Human-like Association and Graph Routing Retrieval

M-Flow focuses on the organizational form of memory and proposes an innovative inverted conical four-layer directed graph structure. Information is stored hierarchically along the "overview-concept-entity-detail" path, enabling highly structured associative activation through graph routing during retrieval. This makes

memory retrieval no longer an isolated similarity match but a hierarchical traversal that conforms to human associative habits. M-Flow represents the structural pinnacle of knowledge networks in semantic memory, providing a solid foundation for semantic resource organization in SOMA.

2.3 Human Multiple Memory Systems and Cognitive Development

Cognitive neuroscience suggests that human memory is not a single system, but rather comprises sensory memory, working memory, and long-term memory encompassing episodic, semantic, and procedural memory. During individual development, humans follow a "generalization-first, specification-later" trajectory—infants initially learn universal principles before gradually developing detailed discrimination abilities. The mechanism of pattern completion (activating complete memory from partial cues) and pattern separation (distinguishing similar but dissimilar memories) in the hippocampus provides a biological basis for tile-driven bidirectional associative retrieval.

However, typical highly intelligent individuals often build a very solid metacognitive framework on top of these systems. This framework consists of a few universal foundational principles (such as first principles, systems thinking, and contradiction analysis) which function like a filter, re-encoding and indexing all internal and external information. This phenomenon reveals that we can implant a similar "thinking framework engine" into AI to serve as the supreme coordinator for memory storage and retrieval.

3. The Sage Thinking Model: Governing Memory Through Frameworks, Nourishing the Present with Resources

The core principles of our refined Sage Thinking Model can be summarized as follows:

1. Problem Decomposition: When encountering any input (problem, task, or scenario), first decompose it into key dimensions, underlying contradictions, and system boundaries through an internal thinking framework. This process relies entirely on the framework's logical reasoning capabilities rather than memory retrieval.

2. Bidirectional Resource Activation: Each decomposed analytical point simultaneously initiates two-way queries to the memory system:

- **Top-down Semantic Query:** Use concepts from the analytical point as keys to locate relevant patterns, facts, and models within the semantic knowledge network.
- **Bottom-up Contextual Association:** The analytical point triggers specific experiences, cases, and scenario fragments related to the concept, forming a rich context.

3. Memory Puzzle Assembly and Solution Synthesis: Activated memory fragments are not fully adopted but ranked based on their "current relevance potential" (weighted by recency, importance, frequency of use, etc.). Only the most relevant "resources" are selected as thinking materials, and combined with framework reasoning, the optimal solution or insight is rapidly synthesized.

4. Reflection and Evolution: After task completion, experiences are reflected upon and refined. Successful action sequences can be solidified into procedural memories (skill modules); new cognition may modify or expand the framework's connection weights or even its underlying patterns. This is "self-growth" – the simultaneous evolution of cognitive frameworks and optimization of memory, making individuals increasingly wise through application.

Memory retrieval in this model is nonlinear and hierarchical: rather than following chronological order or simple keyword matches, it instantly locates information through the "network of meaning" constructed by cognitive frameworks. The distinction between long-term and short-term memory is elevated here to a continuum of "associative potential," determined solely by its relevance to current problems.

4. SOMA System Architecture

Based on this model, the SOMA architecture comprises four core modules: the Thinking Framework Engine, Hierarchical Memory Database, Bidirectional Activation Scheduler, and Metacognitive Evolver. The overall data flow is as follows:

Question input → Decomposed into analytical subgraphs by the Thinking Framework Engine → Bidirectional activation of contextual/semantic memory database → Sort by association potential → Retrieving top-level resources to synthesize solutions → Post-execution metacognitive reflection solidifies skills and updates frameworks.

4.1 Thinking Framework Engine

This serves as SOMA's "intelligent core," storing and operating an evolving set of fundamental rule patterns. Each rule node contains:

- **Rule name** (e.g., "First Principles," "80/20 Rule")
- **Weight:** Represents the current assessment of trust and generality in the rule, which can be adjusted based on experience.

- **Rule relationships:** Define how rules interact to form problem-solving templates.

When a problem is received, the engine selects the most relevant patterns based on their weight scores and generates a series of analytical focal points from the relationship graph, forming a structured decomposition of the problem. For example, analyzing the issue of "new product growth stagnation" may activate "system thinking" and "contradiction analysis," yielding focal questions such as "What are the key enhancement loops in the system?" and "Where lies the core contradiction?"

4.2 Hierarchical Memory Database: Dual Storage for Narrative and Semantic Data

Memory units share a unified base class but fall into two categories:

- **Narrative Memory:** Stores complete event records with contextual tags and timestamps, preserving spatiotemporal context.
- **Semantic Memory:** Stores abstract triadic knowledge (subject, predicate, object) with confidence scores, forming a semantic network similar to M-Flow (implementable using graph databases or vector libraries).

Each memory unit carries importance scores, timestamps, and retrieval counts to calculate key metrics for subsequent sections.

4.3 Bidirectional Activation and Association Potential Calculation

This is the core scheduling algorithm enabling "living in the present with resources." For each analytical focal point generated by the framework engine, the scheduler executes concurrently:

1. Semantic Triggering: Identifies knowledge triples matching the focal keywords within the semantic network.

2. Narrative Association: Search the narrative database for events containing the focal concept in context or content.

3. Relevance Potential Score: Calculate the "current relevance" for each retrieved memory unit:

$$R(m) = \frac{1}{1 + e^{-k \cdot (I + \alpha F)}} \cdot \frac{1}{1 + \delta t}$$

Where I represents memory importance, F is the normalized cumulative frequency of retrieval, δt denotes the time interval (in days), and k, α are tuning parameters. This formula ensures that important, frequently used, and recent memories receive higher scores while avoiding over-reliance on a single dimension.

4. Top-K Selection: All retrieved memories are ranked in descending order of $R(m)$, and the top K are selected as "current knowledge base" and fed into the solution synthesis module (typically an LLM that generates final responses or action plans based on this knowledge and framework prompts).

This scheduling process is entirely problem-centered, non-linearly bridging narrative and semantic dimensions as well as long-term and short-term contexts.

4.4. Metacognitive Evolution and Skill Solidification:

Upon task completion, SOMA enters a reflection phase:

- If the solution proves effective, the sequence of problem-solving steps is encapsulated into a skill pattern and stored in the procedural memory repository (similar to Gene solidification in EvoMap). Subsequent encounters with similar problems allow direct reuse without re-examination, significantly reducing computational overhead.
- Simultaneously, the weight of successfully applied cognitive patterns is modestly increased; patterns long inactive gradually lose weight. Advanced evolution occurs through autonomous induction of new patterns: as the system accumulates numerous successful cases, it identifies novel universal patterns via clustering and abstraction, automatically integrating them into the framework to achieve true "self-growth."

5. Core Algorithm and Code Abstraction

The following Python pseudocode demonstrates the main workflow of the SOMA scheduling core, embodying the concept of integrating retrieval and memory.

```
```Python
Class SOMA_Core:
 def __init__(self):
 self.framework = WisdomFramework () # Thinking Framework Engine
 self.episodic = EpisodicStore () # Plot Memory Store
 self.semantic = SemanticGraph () # Semantic Knowledge Graph
 self(skill = {})

 def respond_to (self, problem: str) -> str:
 # 1. Decompose
 foci = self.framework.decompose (problem)

 # 2. Bidirectionally activate and merge
 candidates = []
```

```

for focus in foci:
 candidates += self.semantic.query (focus)
 candidates += self.episodic.query_by_association (focus)
3. Sort by association potential and deduplicate
 sorted_mems = sorted (set (candidates),
 key=lambda m: m.relevance_potential(),
 reverse=True)
4. Select top-level resources and synthesize a solution (LLM can be invoked here)
 top_mems = sorted_mems[:5]
 solution = self.synthesize(problem, foci, top_mems)
5. Update the access footprint for subsequent potential calculation
 for m in top_mems:
 m.access_count += 1
 return solution

def reflect_and_evolve(self, problem, solution_steps, success):
 if success:
 self(skill[hash(problem)] = SkillPattern(steps=solution_steps)
 self-framework.reinforce_laws_used(problem)
 ...

```

In the engineering implementation, the WisdomFramework maintains a weighted directed graph internally, and the decompose method generates analysis paths through graph search. Semantic graphs can be implemented using NetworkX or Neo4j, while narrative storage employs metadata-supported vector databases. The entire system seamlessly integrates with frameworks like LangChain, utilizing `respond_to` as the top-level decision-making tool for Agents.

## **6. Application Scenarios:**

### **Digital Avatar and Think Tank Management**

SOMA is inherently designed as the intelligent brain for personal digital avatars. The owner's diaries, articles, and project documents are integrated as narrative memories, while reading notes and professional knowledge are converted into semantic triples. The cognitive framework, customized according to the user's philosophical beliefs, serves as the avatar's "cognitive gene." When interacting with the avatar or assigning tasks, SOMA analyzes problems uniquely, draws upon the user's accumulated expertise, and delivers highly personalized and insightful responses—far surpassing the general-purpose models.

Within think tank management systems, SOMA restructures vast amounts of unstructured information, navigating researchers directly to the most relevant deep connections for their current research, enabling instant knowledge reorganization. Its metacognitive evolution mechanism ensures the system increasingly aligns with users' cognitive patterns, fostering genuine knowledge co-creation.

## **7. Discussion:**

### **The Paradigm Shift from "Memorization" to "Insight"**

SOMA's design philosophy represents a fundamental paradigm shift: the benchmark for evaluating memory systems is no longer storage capacity or retrieval speed, but rather the "activability" of memory and the "depth" of problem-solving. By endowing memory with a "meaning index" through its cognitive framework, SOMA enables even edge devices with limited computing power to demonstrate exceptional intelligence through efficient resource allocation—much like how a wise individual

does not require a supercomputer's massive brain. This provides novel approaches for large-scale model applications in low-resource environments.

Future research will focus on: fully automated evolutionary algorithms for the framework, sharing and collision mechanisms among multi-Agent cognitive frameworks, and the integration of a hippocampal-like pattern separation mechanism to optimize semantic memory generalization and discrimination, thereby aligning SOMA's cognitive development more closely with the authentic trajectory of the human brain.

## 8. Conclusion

This paper presents SOMA—an integrated intelligent management system of retrieval and memory. It combines the self-evolving genes of EvoMap and the associative graph structure of M-Flow, while innovatively incorporating the core cognitive model of "using frameworks to govern memory" characteristic of human wisdom. Through decomposition of the cognitive framework engine, bidirectional activation-based resource allocation, and metacognitive evolution, SOMA enables AI Agents to transform past experiences into precise resources for solving current problems like highly intelligent humans, achieving self-growing and intelligent memory systems. SOMA is not merely a technical architecture but also a philosophical practice guiding AI toward true "insight".

## References

(This is a technical preprint; subsequent references provide conceptual sources)

- [1] EvoMap Project: Self-evolving memory and Agent skill genes. GitHub.
- [2] M-Flow Project: Associative memory based on inverted conical graph routing. GitHub.

[3] Kumaran, D., et al. (2016). What Learning Systems Do Intelligent Agents Need? Trends in Cognitive Sciences.

[4] Schacter, D. L. (1996). Searching for Memory. Basic Books.

**In conclusion, I am Sun Yan from Zero Entropy Academy. These reflections stem from my journey from initial AI learning to recent development of the Zero Entropy Think Tank project. I am currently working to implement this system – feel free to explore its feasibility if interested.**