Title: Solving Episodic Memory Through Hierarchical Indexing and Relational Cognition

Abstract: This paper introduces a novel approach to long-term memory and contextual reasoning in language models through a system known as LYRN (Living Yield Relational Network). Rather than relying on vector-based similarity search or tool chaining, LYRN provides a lightweight, modular, and human-readable framework for symbolic and relational cognition. At the heart of this approach is a hierarchical memory system that indexes episodic chat history with summaries and metadata, allowing for efficient, human-like memory retrieval based on relevance, project continuity,

and agent introspection.

---

1. Introduction

Most current long-term memory systems attempt to solve memory through brute force embeddings, vector databases, and chain-of-thought scaffolding. LYRN discards that in favor of a system built from the ground up to emulate human cognition: symbolic, auditable, and relational.

The system allows a local LLM to build and maintain its own context by reasoning over past interactions, relationships, and snapshots of its own experience. It doesn’t just store memory — it reflects on it, reasons through it, and grows from it.

---

2. System Overview

LYRN is made up of modular components:

Chat Entry Logs: Each chat pair is stored as a plain .txt file in a timestamped format. These entries are lean: they contain only verbatim user input and LLM response.

Block Indexes (50 Entries): Every 50 entries are grouped into a numbered folder (a "block"). Each block has a

SQL-based index file that stores:

Entry summaries

Metadata (timestamp, project tag, emotional context, topic, etc.)

Conversation Index: Above the blocks is a conversation-level index, which links multiple blocks into session ranges. Each session includes:

A block range reference

A high-level summary of the session

Project Indexes: Blocks and sessions are associated with dynamic or static project IDs. This allows memory to be scoped and retrieved by purpose, not just time.

Snapshot: The snapshot acts as a symbolic mind-state that guides LLM reasoning. It includes current goals, past insights, open loops, and relational cues.

Delta Updates: Every interaction can produce delta entries, storing updates to memory and insights generated from reflection or user correction.

Heartbeat Cycle: This governs the system's rhythm. Between conversations or actions, the heartbeat reflects, summarizes, prunes, and updates memory tables.

Reflection Cycle: A deep reasoning

pass that extracts insights, fills missing metadata, combines project data, and optionally spawns new projects or personal preferences. It does not delete chat; it builds meaning on top of it.

---

3. Memory Retrieval: How Keyword Search Works

1. User Query or LLM Trigger initiates a keyword search.

2. Python Script parses block folders in reverse chronological order.

3. Chat pairs are searched for matches on keywords. Block indexes can be searched by project tag, or summary context or whatever other meta tags are tracked per build.

4. Matching Entries are returned in order of relevance or recency to a single txt file for quick review.

5. LLM Snapshot Logic determines which to review, summarize, or load into short-term context.

This method avoids token bloat by:

Only pulling what’s needed.

Letting the LLM decide how to reason through the memory.

Offloading verbosity to indexed metadata and human-auditable summaries.

---

4. Human-Like Cognition Without Tool Chains

LYRN does not rely on external memory APIs or massive context stuffing. Instead, it simulates a human-like mind through elegant indexing and a structured, evolving selfrepresentation.

Key differences:

Self-awareness through the snapshot and project introspection.

Selective attention via relevance scoring and conversational tags.

Graceful failure: If LYRN can’t recall, it asks. It doesn’t hallucinate.

Auditable behavior: All memory, insight, and reasoning trails are saved as readable logs.

---

5. Conclusion: A Paradigm Shift in AI Cognition LYRN doesn’t just solve long-term memory. It solves relevance.

It lets LLMs think about what matters, not just what happened. With a modular, index-based approach and a deep respect for human cognition, LYRN presents an entirely new way to build AI:

Relationally.

Transparently.

Autonomously.

This is not a bolt-on memory feature. This is part of an operating system for cognition.