Summary of papers:
HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models
Cognitive Architectures for Language Agents

Peixuan Han (ph16), Xiaocheng Yang (xy61), Zirui Cheng (ziruic4)

Problem and Motivation

Recent years have witnessed the rapid growth of large language models (LLMs) in terms of agentic ability. They have been endowed with natural language comprehension and tool usage abilities to react to human users and the environments and employ proper tools to perform certain tasks. Therefore, a study on language agents (Weng, 2023; Wang et al., 2023b; Xi et al., 2023; Yao and Narasimhan, 2023) has become an emerging trend in the research community.

One trajectory of this study is to take inspiration from neuroscience to solve specific issues with the current implementation. A long history of evolution equips mammalian brains with the crucial ability to store large amounts of world knowledge and continuously integrate new experiences without losing previous ones (Eichenbaum, 2000). Speaking of the memory component of language agents, retrieval-augmented generation (RAG) (Lewis, 2020; Izacard, 2022; Ram 2023; Xie 2024) is the de facto solution for long-term memory. However, vanilla RAG cannot effectively integrate information across different resources. Existing variations of RAG (Press, 2023, Trivedi 2023) fall short of the knowledge integration ability as well. Therefore, it might be promising to mimic the way mammalian brains function in order to solve this problem. In this summary, we will present HippoRAG (Gutiérrez, 2025) as one representative example on this track.

Besides, building high-level conceptual frameworks is also urgently called for. Although researchers use roughly similar components (such as ‘tool use’, ‘grounding’, and ‘actions’) to design agentic processes, there is currently no unified standard or consensus on the terminologies of language agents, which imposes barriers to making comparisons between different works. In this summary, we will present CoALA (Sumers, 2024), which is one recent effort made to that end.

Related Work

Cognitive Architectures of LLM Agents. Cognitive architectures have historically played a significant role in both psychology and computer science, finding applications across diverse domains including robotics (Laird et al., 2012), military simulations (Jones et al., 1999; Tambe et al., 1995), and intelligent tutoring systems (Koedinger et al., 1997). Despite their initial prominence, these architectures have seen declining adoption within the AI community in recent decades. This decline can be attributed to two primary limitations: their restriction to domains describable through logical predicates, and their reliance on extensive pre-specified rules for operation.

Memory Mechanisms of LLM Agents. The conceptualization of memory in cognitive systems draws heavily from psychological theories (Atkinson and Shiffrin, 1968). The primary types include working memory and long-term memory, each serving distinct functions. Working memory, as described by Baddeley and Hitch (1974), maintains the agent's current state, encompassing recent perceptual inputs, active goals, and intermediate reasoning results. Long-term memory is further subdivided into three categories: procedural memory (containing the production system's rules), semantic memory (storing factual world knowledge) (Lindes and Laird, 2016), and episodic memory (recording sequences of past behaviors) (Nuxoll and Laird, 2007). With recent advances of LLMs, working memories for agents can be simply implemented within the context windows. In the meanwhile, LLMs also demonstrably encode substantial world knowledge within their parameters (AlKhamissi et al., 2022; Chen et al., 2024; Geva et al., 2023; Petroni et al., 2019), which might serve as long-term memories.

Current Techniques for Memory Management. Despite various approaches including fine-tuning, model editing (De Cao et al., 2021; Meng et al., 2022; Mitchell et al., 2022), and external parametric memory modules (Park et al., 2024; Wang et al., 2024), a robust solution for continual learning in LLMs remains elusive. Recent advances in Large Language Models (LLMs) have introduced novel approaches to augmenting memory systems. These implementations can be categorized into both retrieval-augmented generation (RAG) and long-context window methods. RAG methods offer a more flexible approach to long-term memory implementation (Izacard et al., 2022; Lewis et al., 2020). Advanced RAG systems can integrate information across multiple knowledge elements through multi-step retrieval and generation processes (Jiang et al., 2023; Press et al., 2022; Shao et al., 2023; Trivedi et al., 2023). In the meantime, recent developments have led to dramatic increases in context lengths for both open and closed source LLMs (Chen et al., 2023; Ding et al., 2024; Fu et al., 2024; Peng et al., 2023). Both methods have significantly improved LLMs' capabilities in memory management.

Solution Overview

HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

1. **Indexing.** This is the process of building the retrieval information database. For each passage, this work uses an instruction-tuned LLM to conduct named entity recognition (NER) and triplet extraction (IE). Specifically, they first extract a set of named entities from each passage and then add the named entities to the OpenIE prompt to extract the final triples, which also contain concepts (noun phrases) beyond named entities. The triplets are edges in the knowledge graph, and the extracted entities are nodes. In addition, an encoder model is used to find synonymy relations, adding extra edges to the KG. Finally, they record a matrix P indicating the occurrence of each entity in each passage.
2. **Retrieval.** During retrieval, NER is also applied to the query to find the entities that appear in the query. The encoder model is then used to find the most relevant entities in the KG as key entities. Then, the method conducts PPR on the knowledge graph to decide the importance of each node. (PPR: random walk on the graph; for

each step, either move to an adjacent node or return to one of the key nodes. The importance of node is defined as the occurrence frequency of each node after enough steps.) The importance vector is multiplied by P, generating the importance of each passage. Top-k passages are returned as retrieval results.
3. **Node Specificity.** When a node occurs in a lot of passages, its importance should be reduced. A factor s=1/p (p: the number of occurrences of a node in all passages) is multiplied by the node importance during retrieval.

In conclusion, this paper proposes a novel RAG framework by storing entities in a knowledge graph, therefore achieving retrieving multi-hop relationships in a single retrieval.

Cognitive Architectures for Language Agents

The paper proposes **CoALA (Cognitive Architectures for Language Agents)**, a framework to systematically design and analyze language agents powered by large language models (LLMs). By unifying memory management, action selection, and structured reasoning, CoALA aims to bridge symbolic AI principles with LLM flexibility, enabling systematic agent design and comparison. The key components in CoALA include:

1. **Memory**: Modular storage with *working memory* (temporary state) and *long-term memory* (episodic, semantic, procedural knowledge).
2. **Action Space**: Structured into *external actions* (interacting with environments) and *internal actions* (reasoning, retrieval, learning).
3. **Decision-Making**: A cyclic process with *planning* (proposing/evaluating actions via LLM-based reasoning) and *execution* (grounding actions or updating memory).

Limitations

HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

1. The models used in this work did not go through extra training. Components like NER and OpenIE are directly used off the shelf, leading to error cases in the graph formation step. Notably, there is an inconsistency between OpenIE performance on long documents and short documents.
2. This work adopts the PPR algorithm, which is a trivial solution to searching important nodes given a knowledge graph. However, the error cases in the graph search step indicate that PPR is still not reliable enough.
3. Although HippoRAG using Llama-3.1 can achieve similar performance to closed-source models, the efficiency and efficacy as the size of the document set grows are still unclear.

Cognitive Architectures for Language Agents

1. It is unclear whether reasoning should be multi-modal.
2. It is hard to define what is external to an agent and what is internal to an agent.
3. The agent framework designed for digital environments might not be sufficient for the physical world due to the differences between the two.
4. There's no agreement on whether learning should also be a part of action.
5. How more powerful LLMs will change the agent design is still unclear. It is possible that as models gain greater capacity, many of the framework's components will be unnecessary.

Future Research Directions

1. Finetune model for better named entity recognition and information extraction performance to improve HippoRAG.
2. Integrate better graph search algorithm to mitigate graph search errors in HippoRAG.
3. Go beyond the form of knowledge graph by extending the triplet format or performing operations on other flexible structures of information instead.
4. Explore the possibility of multi-modal reasoning.
5. Experiment with agents designed for the physical world.
6. Enhance model abilities to simply the agent framework.

Summary of Class Discussion

Q: What is the main novelty of the paper HippoRAG?
A: Although KG is commonly used in retrieval, this paper proposes a method to build a KG with passage, and uses PPR on the KG to decide which passage to retrieve.
Q: How may the relation type in KG edges be better used to decide the importance of each node?
A: 1) we can add weight to each edge based on the strength of the relation. 2) the traditional IE triplet (entity, relation, target entity) isn't flexible enough to express complex relationships. Maybe we need a novel way of expressing external memories.
Q: Are the components mentioned in CoALA implemented in real systems?
A: As a survey, the CoALA paper shows a "model" that combines different effective approaches for building LLM memory. Which one(s) to use depends on the actual scenario.
Q: How should language agents use different types of memories?
A: In previous studies, researchers have proposed some weight-based methods to enable agents to utilize different types of memories (e.g., work memory, procedural memory, semantic memory, and episodic memory) to improve their performances. However, there is still exploration space for optimization.