



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



SURUTHI S · Follow

6 min read · Jul 6, 2024



Listen

HippoRAG is a novel retrieval framework, inspired by hippocampal indexing of the Human Brain to retain Long term memory and integrate information from multiple parts of the document. It uses Knowledge graph with Personalized PageRank Algorithm allowing for more accurate retrieval results and ultimately mimicking the role of neocortex in storing human memory.

Overview of RAG

Retrieval augmented generation (RAG) is a widely used framework for **integrating external data** and giving LLM **additional context** to improve the quality of information retrieval in domain-specific areas. The documents are converted to chunks, and vector representation are created using embedding techniques. These vector representations are then stored in Vector Databases (Vector DBs).

When a query is made, the relevant vectors are retrieved from the Vector DB based on their **similarity to the query**. These retrieved chunks of information are then passed as context to the LLM. This allows the LLM to generate *more accurate and*

contextually relevant responses by leveraging the specific information contained within the external documents.

Challenges in Current RAG Systems:

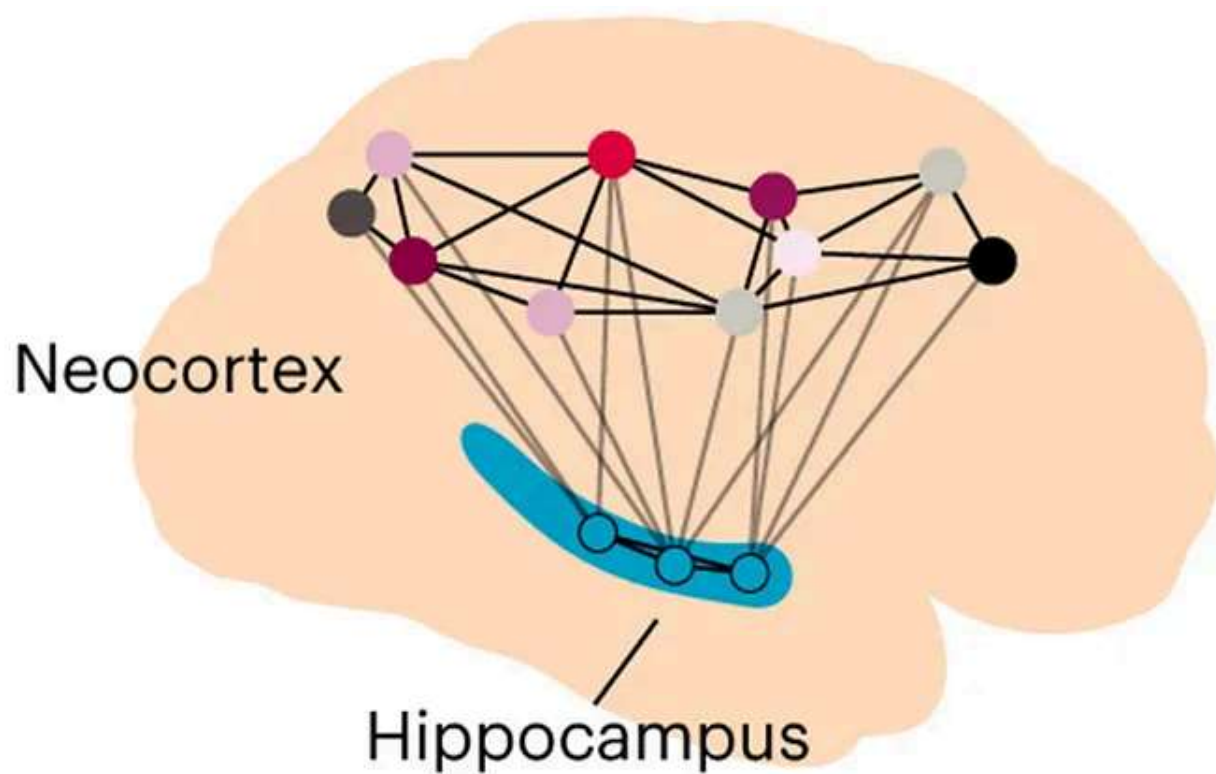
In terms of **Multi-hop Question Answering** (the answer for the user question consists of different fragments located across different documents), present RAG systems have a lack of ability to generate adequate responses. It uses **Multiple Retrievals** and LLM Generation iteratively to **concatenate different pieces of information**. However, even a perfectly implemented multi-step RAG can be insufficient to achieve numerous levels of knowledge integration. This is because the passages are encoded in isolation.

Many challenging domains, like **scientific literature review**, **legal case briefing**, and **medical diagnosis**, acquire context by integrating the content across passages or documents. Current RAG systems seem not to be a one-fit solution for these integration tasks; this is where **the HippoRAG comes into action**.

HippoRAG is inspired by the **Hippocampal Indexing Theory**, which states that the hippocampus (a C-shaped structure located in the medial temporal lobe and part of the limbic system) acts as **an index or pointer** to the locations of memory traces stored in the neocortex.

Mechanism:

- **Encoding:** *During the encoding of new information, the hippocampus forms a unique index that represents the new memory.*
- **Storage:** The actual memory traces (sensory, contextual, and conceptual information) are distributed across *different regions of the neocortex*.
- **Retrieval:** When a memory needs to be retrieved, *the hippocampus uses the index to reactivate the distributed neocortical patterns*, bringing the memory back to consciousness.



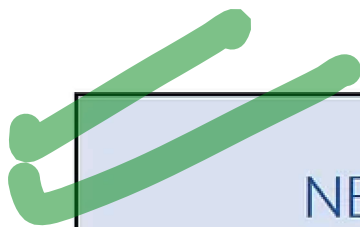
Hippocampus creates index for the memories to be stored in different part of neocortex

The functions of the following part are taken as inspiration to develop the HippoRAG:

1. **Neocortex** — processing and storing memory
2. **Hippocampus** — indexing the memory stored in the neocortex
3. **Para-Hippocampal Region (PHR)** — forms a pipeline between neocortex and hippocampus

Applying the same methodology to the RAG part:

1. **LLM** acts as an artificial neo-cortex processing and extracting high level information.
2. **Knowledge Graph with Personalized PageRank (PPR)** as Hippocampus
3. **PHR** as Retrieval encoders, being middle in the pipeline for dense encoders fine-tuned for the retrieval. They determine the similarity and synonymy among nodes



NEOCORTEX	LLM
HIPPOCAMPUS	Knowledge Graph + PPR
Para-hippocampal Regions	Retrieval Encoders

This paper is centered around the **Hippocampal Memory Indexing Theory**, where Teyler and Discenna propose that human long-term memory is composed of **three components** (neocortex, hippocampus, and para-hippocampal regions) that work together to accomplish two main objectives: **pattern separation**, which ensures that the representations of various perceptual experiences are distinctive, and **pattern completion**, which allows the recall of full memories from partial stimuli.

This happens in two steps:

1. **Memory Encoding: Memory encoding allows for pattern separation.** High level manipulatable Perceptual stimuli are received and processed in the neocortex, which then passes through para-hippocampal regions to be indexed by the hippocampus.
2. **Memory Retrieval:** Every time partial perceptual stimuli related to previously recorded memory traces are delivered from the PHR pipeline, pattern completion drives memory retrieval subsequent to memory encoding.

The same two steps can be formulated as follows:

The corpora are stored as entities in a knowledge graph (**Offline Indexing**) and retrieved whenever user queries are passed. (**Online Retrieval**)

The indexes created by the hippocampus are prominent and high-level, having mutual connections with each other.

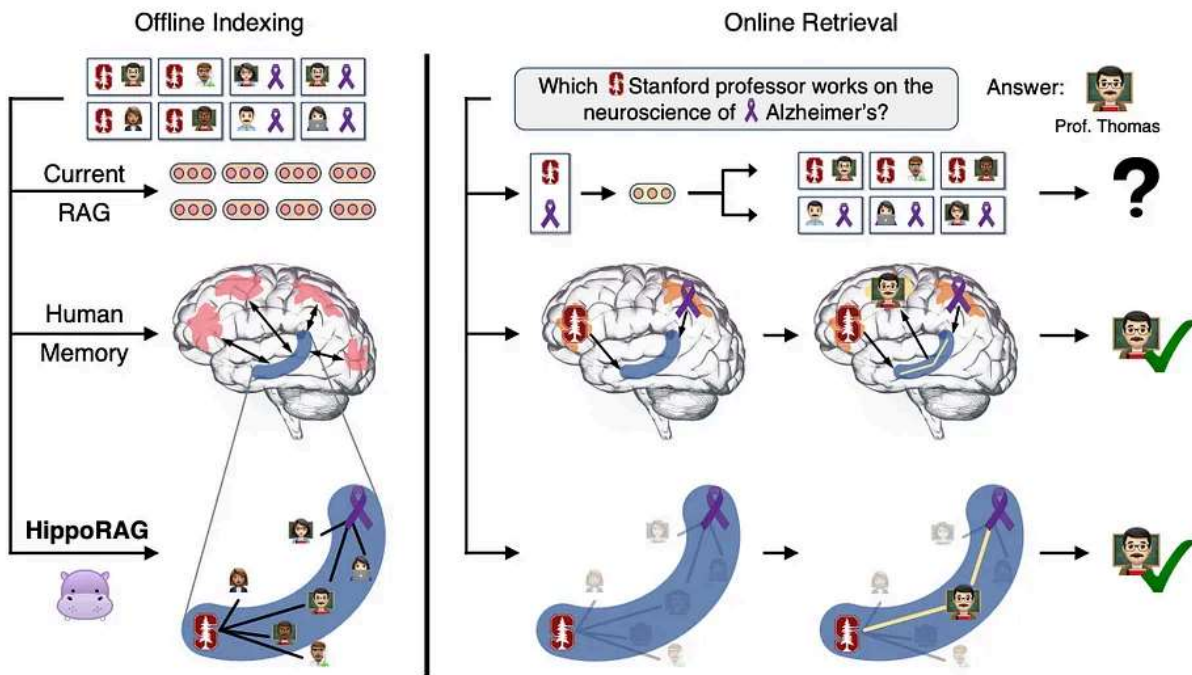


Figure 1: **Knowledge Integration & RAG.** Tasks that require knowledge integration are particularly challenging for current RAG systems. In the above example, we want to find a *Stanford* professor that does *Alzheimer's* research from a pool of passages describing potentially thousands *Stanford* professors and *Alzheimer's* researchers. Since current methods encode passages in isolation, they would struggle to identify *Prof. Thomas* unless a passage mentions both characteristics at once. In contrast, most people familiar with this professor would remember him quickly due to our brain's associative memory capabilities, thought to be driven by the index structure depicted in the C-shaped hippocampus above (in blue). Inspired by this mechanism, **HippoRAG** allows LLMs to build and leverage a similar graph of associations to tackle knowledge integration tasks.

Let us analyze it in a detailed fashion:

Offline Indexing:

1. The instruction-tuned LLM in the initial step of the pipeline handles the input documents by running **named entity recognition** and extracting the entities, which are then segregated into **triples**. This process is known as Open Information Extraction (Open IE).
2. The entities generated are stored in the **Schemaless Knowledge Graph**, which arranges these entities based on their connections. *The triples are discrete noun phrases rather than dense vector representation*; this is the key factor in this approach as it allows more fine-grained pattern separation.

One-Shot Demonstration:

Paragraph:

...

Radio City

Radio City is India's first private FM radio station and was started on 3 July 2001. It plays Hindi, English and regional songs. Radio City recently forayed into New Media in May 2008 with the launch of a music portal - PlanetRadiocity.com that offers music related news, videos, songs, and other music-related features.

...

```
{"named_entities": ["Radio City", "India", "3 July 2001", "Hindi", "English", "May 2008",  
"PlanetRadiocity.com"]}
```

```
{"triples":
```

```
[
```

```
  ["Radio City", "located in", "India"],
```

```
  ["Radio City", "is", "private FM radio station"],
```

```
  ["Radio City", "started on", "3 July 2001"],
```

```
  ["Radio City", "plays songs in", "Hindi"], ["Radio
```

```
City", "plays songs in", "English"], ["Radio City",
```

```
  "forayed into", "New Media"],
```

```
  ["Radio City", "launched", "PlanetRadiocity.com"],
```

```
  ["PlanetRadiocity.com", "launched in", "May 2008"],
```

```
  ["PlanetRadiocity.com", "is", "music portal"],
```

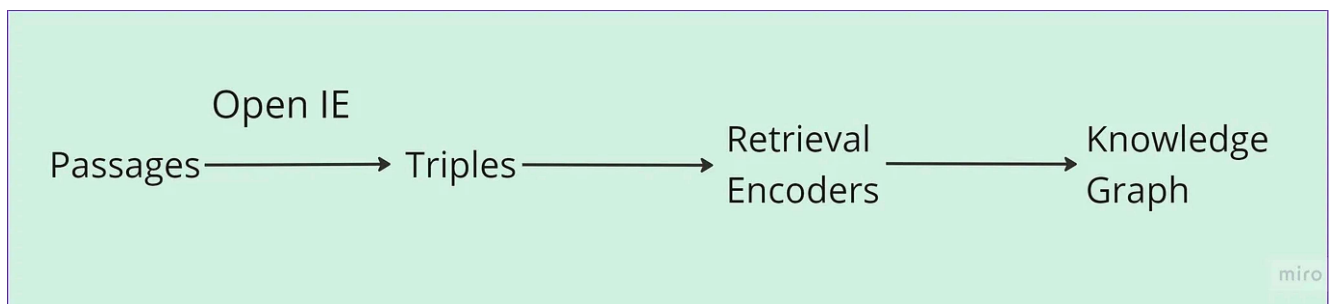
```
  ["PlanetRadiocity.com", "offers", "news"], ["PlanetRadiocity.com",
```

```
  "offers", "videos"], ["PlanetRadiocity.com", "offers", "songs"]
```

```
]
```

```
}
```

Extracting Triples from the passage using Open IE



Online Retrieval:

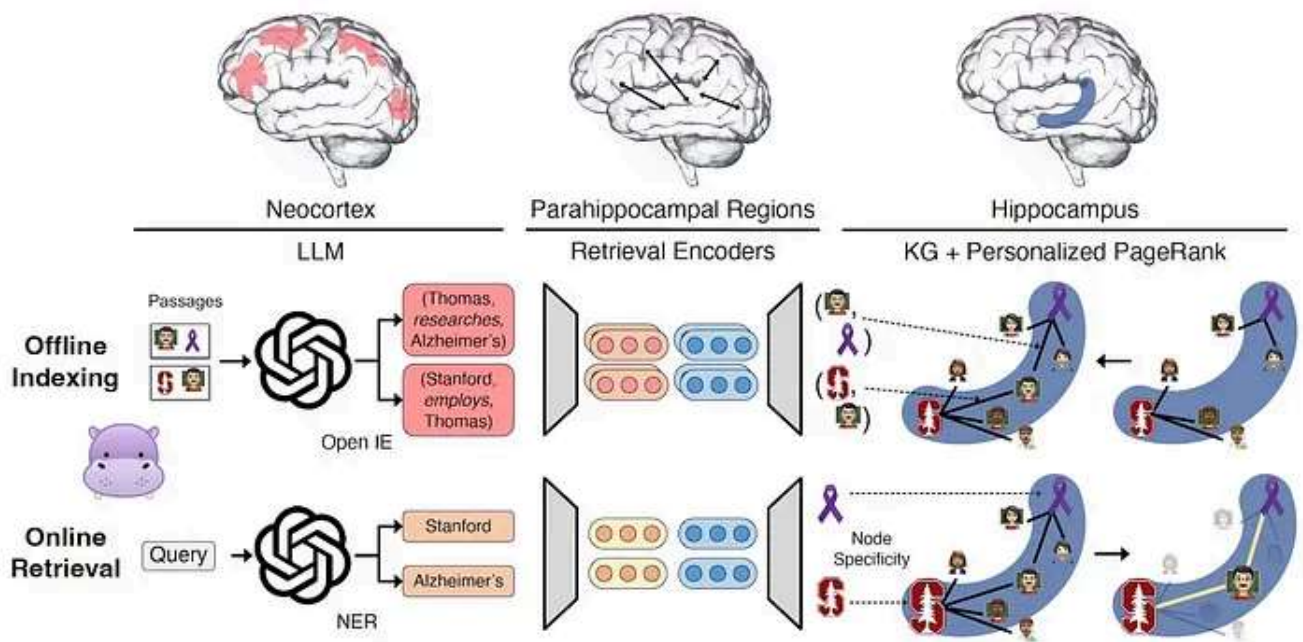
So, whenever a user queries the system,

1. We prompt LLM using a **1-shot prompt** to extract a set of named entities from a query q . The named entities are extracted and selected from the previously defined query named entities (C_q) from the offline indexing.
2. The query nodes are chosen as the set of nodes in N with the highest cosine similarity to the query named entities C_q .
3. After the query nodes are found, we run Personalized PageRank (PPR) over the knowledge graph. The PPR makes sure that each query node has an equal

probability and all other nodes have a probability of zero.

The Personalised PageRank (PPR) algorithm, a version of PageRank that distributes probability across a graph only through a set of user-defined source nodes. This constraint allows us to bias the PPR output only towards the set of query nodes

Consider an example.



Offline indexing:

1. Entities are segregated, and triples are formed eg:(Sarah, researches, Alzheimer's), (Thomas, researches, Alzheimer's), (Stanford, employs, Mike), (Stanford, employs, Thomas) etc....
2. They are processed through retrieval encoders for representation to be stored in the knowledge graph.

Online Retrieval:

1. When the user queries, **"Which Stanford professor works on the neuroscience of Alzheimer's?"**. The named entities are extracted here (Stanford, Alzheimer's) using 1-shot prompting.
2. These named entities (Stanford, Alzheimer's) are then linked to nodes in our KG based on the similarity determined by retrieval encoders. **Once the query nodes are chosen, they become the partial cues from which our synthetic hippocampus performs pattern completion.**

3. PPR is performed for the query nodes; aggregate the output PPR node probability over the previously indexed passages and use that to rank them for retrieval and returns (**Thomas**)

Additionally, to increase the retrieval relevancy, this paper makes use of a mechanism known as **node specificity**.

Global signals for word importance, such as **inverse document frequency (IDF)**, are known to enhance information retrieval. **IDF would be complicated** as it is a global property and triggers all nodes in the hippocampal index every time retrieval occurs.

To overcome this issue, **HippoRAG uses node specificity as an alternative to IDF**, which focuses only on **local signals**. The node specificity of node i is calculated as follows : $s_i = |P_i|^{-1}$, where P_i is the set of passages from which node i was extracted.

Node specificity is used in retrieval by multiplying each query node probability with s_i before PPR; this allows us to modulate each of their neighborhood's probabilities as well as their own.

Some of the prominent features of this approach are:

1. Using knowledge graphs and retrieval encoders to link named entities,
2. Leveraging PPR for ranking passages based on query nodes
3. Using salient noun phrases for the encoding
4. Node specificity

These are the intriguing features that streamline **long-term memory handling and multi-hop question answering**, making the retrieval process more efficient and accurate. This approach enhances the overall performance of the synthetic hippocampus in pattern completion and information retrieval tasks.

A major advantage of HippoRAG over conventional RAG methods in multi-hop QA is **its ability to perform multi-hop retrieval in a single step**.

Retrieval Augmented

Llm

NLP

Langchain

Retrieval Augmented Gen



Follow

Written by SURUTHI S

6 Followers · 24 Following

LLM Enthusiast with a passion for advanced language models and AI-driven solutions. Dedicated to continuous learning and knowledge sharing in the AI community.

No responses yet



What are your thoughts?

Respond

More from SURUTHI S



SURUTHI S

Integrating Computer Vision with Natural Language Processing for Image Understanding

Image Understanding

Aug 14, 2023 🖱️ 104



SURUTHI S

From Local to Distributed: Understanding the Evolution of Version Control Systems

Contents

Demystifying the Tasks of Computer Vision

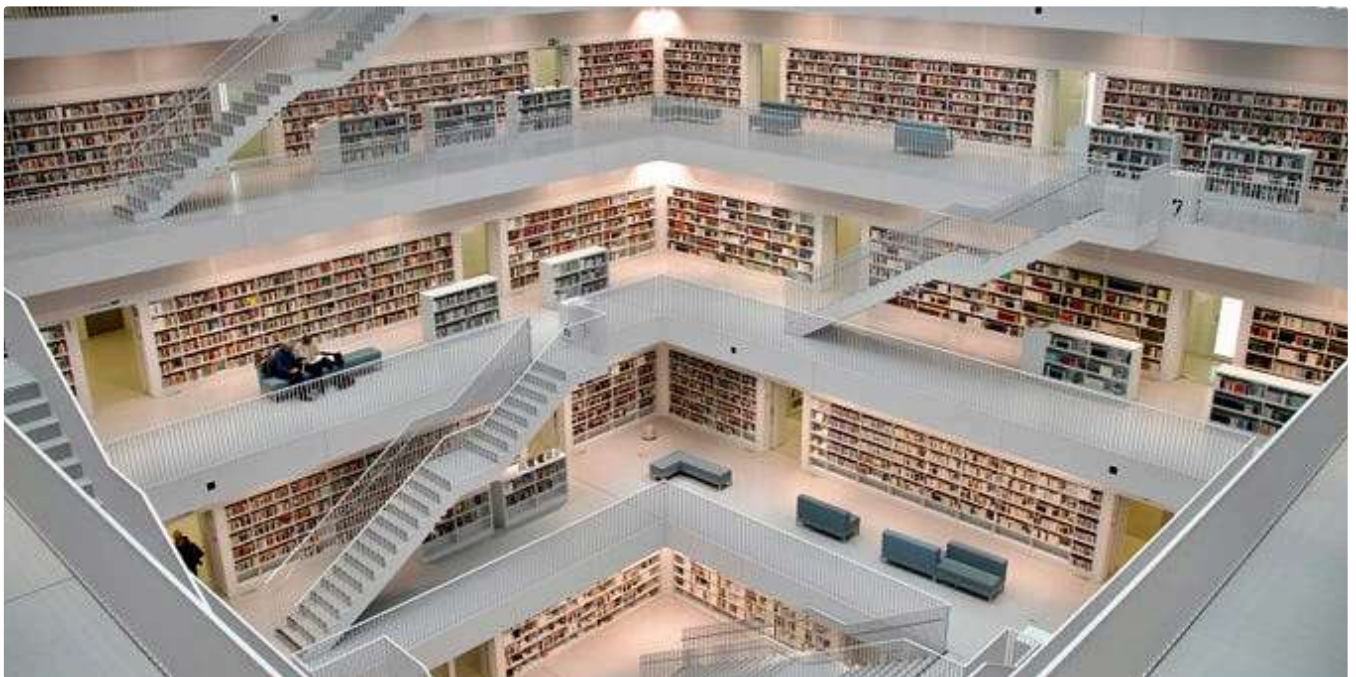
Introduction

Aug 12, 2023 🖱️ 21



See all from SURUTHI S

Recommended from Medium



 In Towards Data Science by Thuwarakesh Murallie

How to Build a Knowledge Graph in Minutes (And Make It Enterprise-Ready)

I tried and failed creating one—but it was when LLMs were not a thing!

🌟 3d ago 🖱️ 517 💬 5





AI In Artificial Intelligence in Plain English by Sarayavalasaravikiran

KAG: A Better Alternative to RAG for Domain-Specific Knowledge Applications

The rise of large language models (LLMs) has brought remarkable breakthroughs in natural language processing (NLP). Retrieval-Augmented...

★ Jan 4 🖱 252 💬 15



Lists



Natural Language Processing

1888 stories · 1544 saves



The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 540 saves



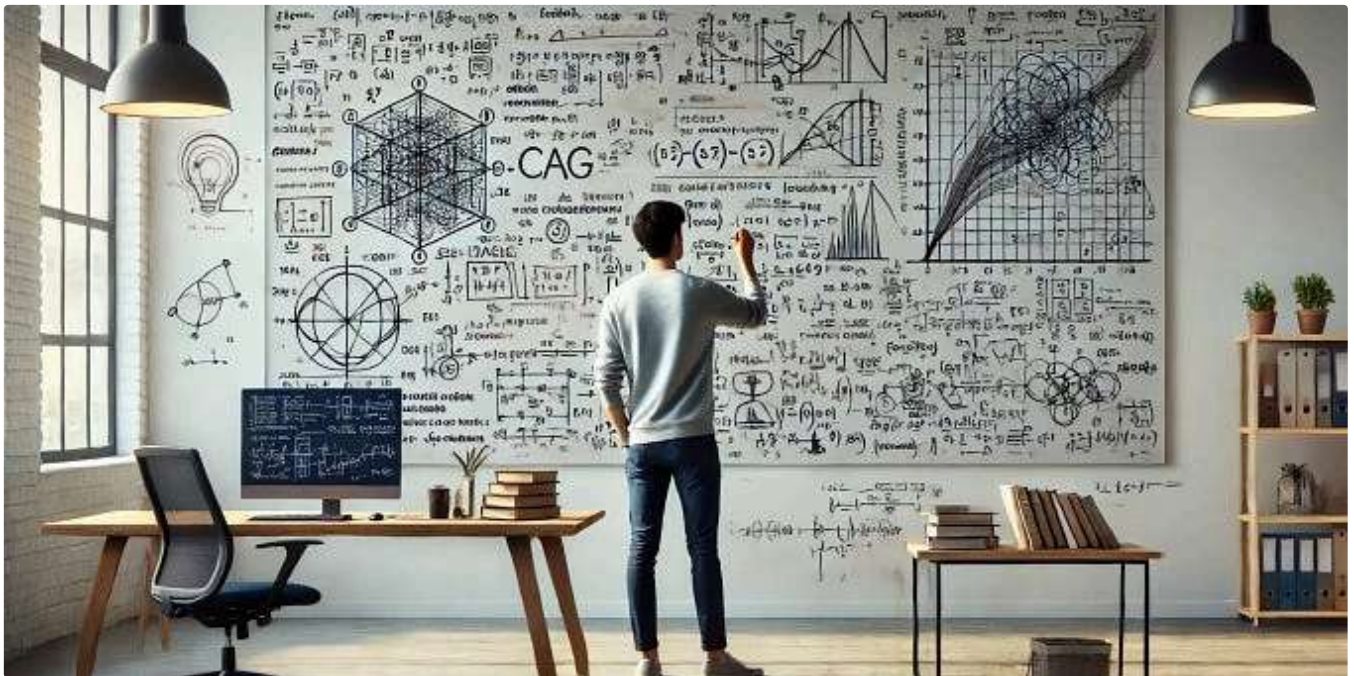
data science and AI

40 stories · 318 saves



ChatGPT prompts

51 stories · 2472 saves





 In Level Up Coding by Dr. Ashish Bamania 

Cache-Augmented Generation (CAG) Is Here To Replace RAG

A deep dive into how a novel technique called Cache-Augmented Generation (CAG) works and reduces/ eliminates the need for RAG.

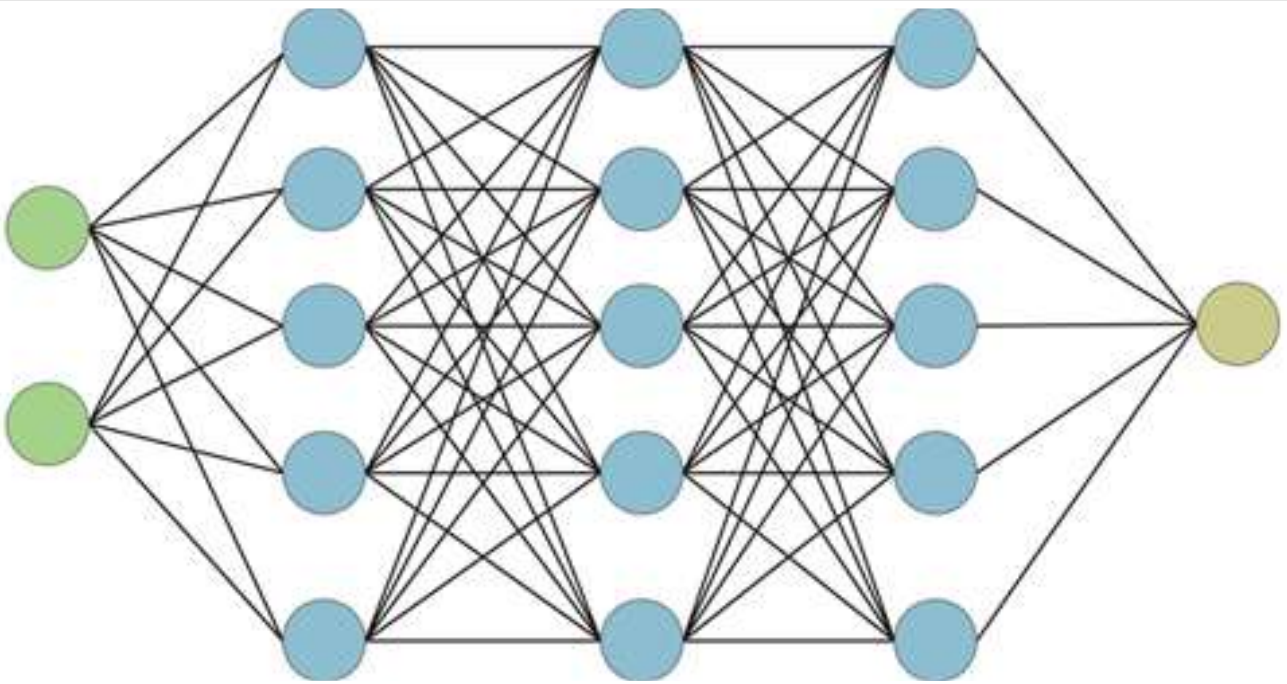
★ Jan 10 🖱 230 💬 8



 In Data Science in your pocket by Mehul Gupta 

Sky-T1-32B-Preview : Open-sourced LLM outperforms OpenAI-o1

UC Berkley's Sky-T1-32B-Preview details

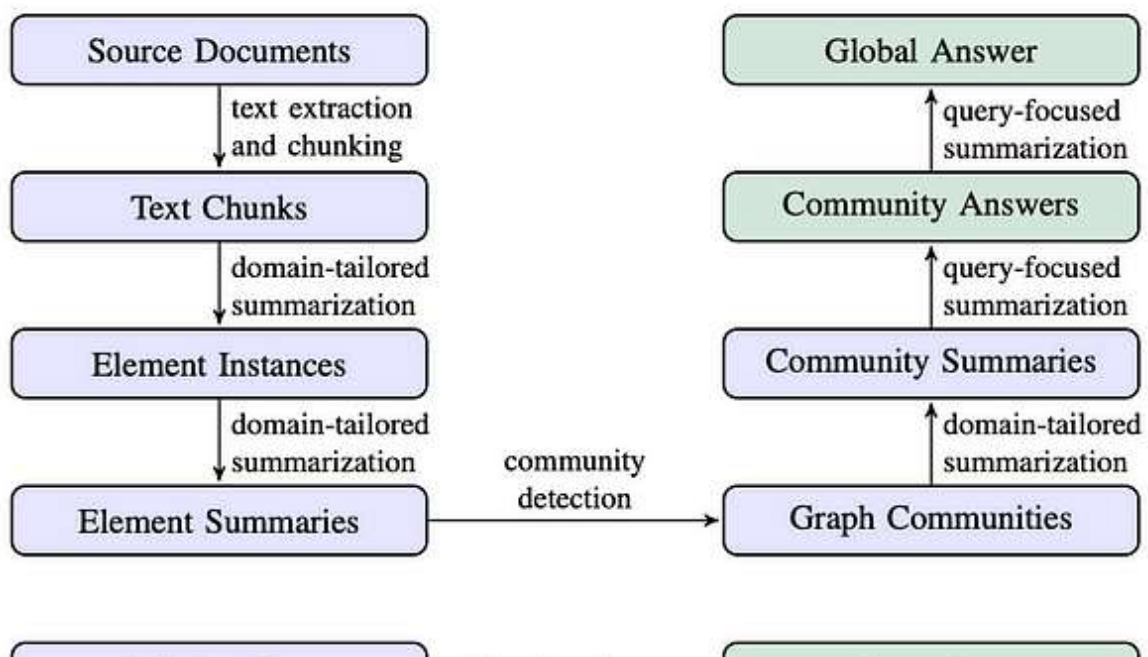


 Adam On Projects 

ChatGPT Research #3: Project Management Use Cases of ChatGPT (Part 1)

This article unpacks the results of the explicit questions I asked on the actual use of ChatGPT.

★ Aug 13, 2024 🖱 6



 Zilliz

GraphRAG Explained: Enhancing RAG with Knowledge Graphs

Introduction to RAG and Its Challenges

Aug 7, 2024  255  3



See more recommendations