Member-only story

LLM Apps : Why Knowledge Graphs are super critical to know if you care about RAG : 1

Aniket Hingane | Day Manager, Night Coder · Follow
5 min read · Feb 21, 2024

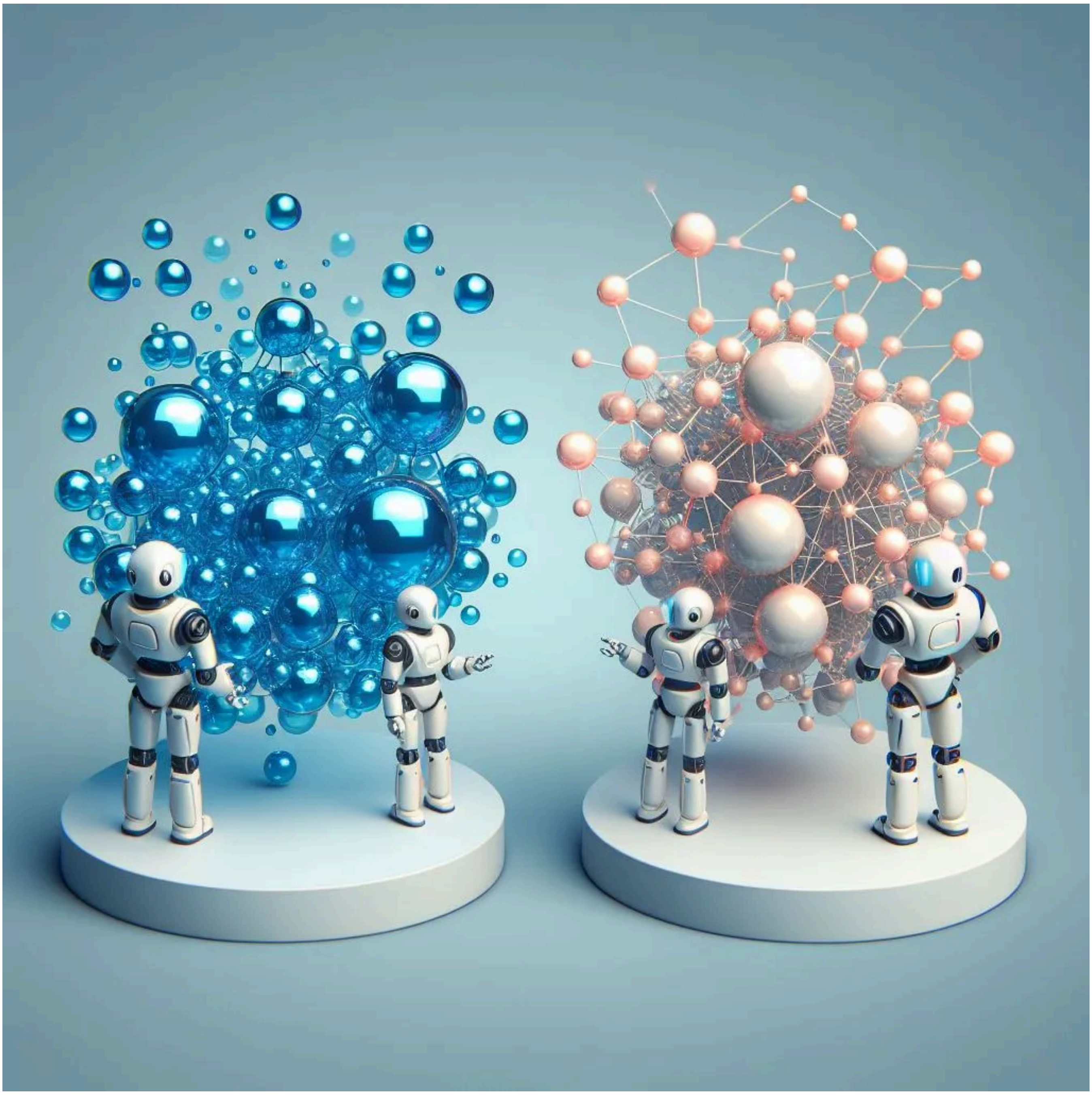
AI Apps Need RAG, RAG Need Knowledge Graph ! Next part we will code RAG on KG.

Why should we care about KG, if Vector DB are in use in RAG?

Retrieval Augmented Generation (RAG) is revolutionizing the way we interact with large knowledge bases. By combining the retrieval of relevant information with the generative capabilities of language models, RAG systems can provide detailed answers, create summaries, and even translate between languages. Yet, the true potential of RAG hinges on how the underlying knowledge is structured. While vector databases have been the conventional choice in many RAG systems, their limitations become clear in applications that **demand deep, sophisticated reasoning**.

This is where knowledge graphs emerge as transformative technology, offering a powerful model for understanding complex, interconnected data.

When is KG better than Vector in RAG?



Isolated Textual Representation

Vector databases represent documents or passages as numerical vectors in a high-dimensional space. The focus is on semantic similarity: documents with similar concepts will be positioned closer together. While this is useful for tasks like fuzzy search, it breaks down when the RAG system needs to understand the nature of relationships between different pieces of information.

Is that really a problem in real world ?

Imagine a RAG system used by a market research analyst. The task is to generate a comprehensive report on a competitor, drawing information from news articles, earnings reports, and industry analyses. A vector database approach might retrieve relevant documents, but it would struggle to piece together the timeline of acquisitions, leadership changes, and evolving product lines — the analyst would be left doing most of the time-consuming connective work.

Lack of Deep Relationships

Vector similarity can tell you *that* two entities are conceptually related, but it won't easily tell you *how* they are related. Are they competitors? Is one a subsidiary of the other? Did they collaborate on a project? These nuanced connections are essential for RAG systems to reason about complex domains.

Is that really a problem in real world ?

Consider a healthcare setting. A RAG system assisting with differential diagnosis needs more than just finding documents containing symptoms and diseases. It needs to understand the hierarchical relationships of medical conditions, causal links between test results and diagnoses, and potential drug interactions. A vector database cannot inherently provide this structural understanding.

Difficulty with Time-Based Reasoning

Vector stores generally represent a snapshot in time. Unless time information is carefully encoded into the vector embeddings themselves, temporal reasoning becomes very challenging. Tracking changes, analyzing trends, and understanding historical sequences are crucial for many RAG applications.

Is that really a problem in real world ?

In financial analysis, understanding the evolution of a company's financial performance over time is key, and involves connecting data points across various reports. A RAG system relying solely on a vector store would find it difficult to answer a question like "How has the company's debt-to-equity ratio changed over the past five years, and what major events might have influenced this change?"

How does KG address these limitations?

Interconnected Data Model

The core strength of knowledge graphs lies in their ability to model knowledge as a network of entities (nodes) and relationships (edges). This mirrors the way we naturally think about the world. For example, a knowledge graph for a market research analyst might include entities like companies, products, executives, and market trends, with relationships like "competes with," "acquired by," "CEO of," and "influences."

A RAG system can then traverse these relationships to uncover information that wouldn't be readily apparent from a collection of isolated documents.

So, how it solves Challenge of Isolated Representation?

With a knowledge graph, the RAG system can follow the chain of connections to understand the full story of a competitor — its acquisitions, leadership changes, evolving product focus — allowing for a comprehensive report surpassing what could be inferred from individual documents within a vector store.

Incorporating Time

Knowledge graphs easily accommodate time-related information. Relationships themselves can have timestamps or durations. Questions about change over time become natural to query.

So, how it solves Challenge of Time-Based Reasoning?

Analyzing a company's financial performance now becomes possible with queries like: "Show me all acquisitions made in the last two years" or "Find news articles discussing declining revenue trends over the past three quarters." The graph structure itself can encode the temporal context.

Explainability and Transparency

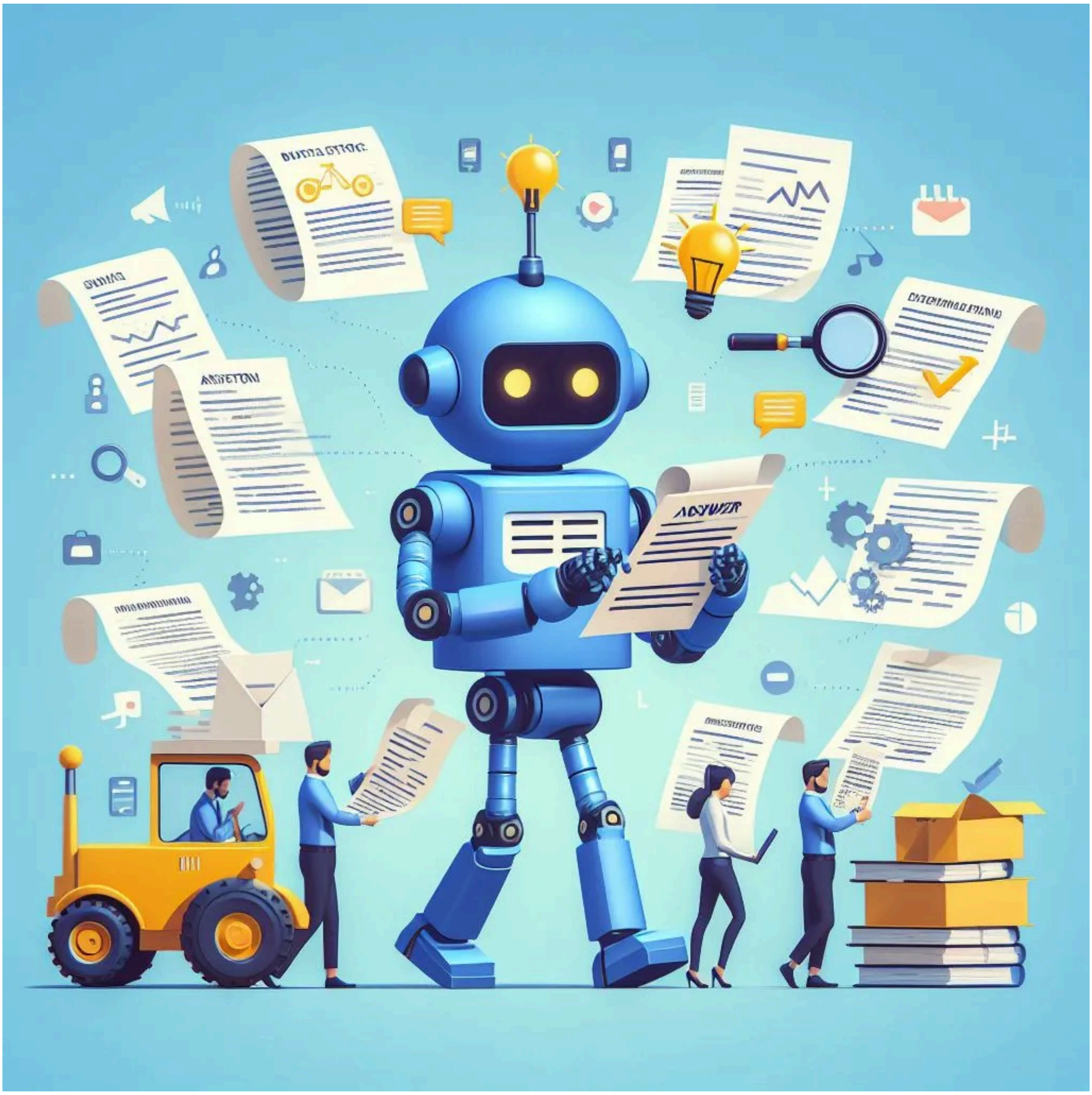
Knowledge graphs, by their very nature, make the reasoning process of a RAG system visible. Since answers are generated by traversing relationships, it's easy to trace *why* and *how* the system arrived at a specific conclusion.

So, how it solves Challenge of Lack of Deep Relationships?

In the healthcare scenario, the knowledge graph makes it clear whether a particular symptom is a common side effect of a drug, a potential indicator of a disease, or strongly linked to a specific test result. This transparency is crucial for building trust in the system's recommendations, especially in high-stakes domains.

> Below one I like most

How Knowledge Graphs Enable Multi-Hop Reasoning



The ability to traverse relationships across multiple “hops” is where knowledge graphs truly shine. Consider this example:

Question: “What are the potential environmental implications of Company X acquiring Company Y??

A knowledge graph-powered RAG system might follow a path like this:

1. **Company X** -> [acquires] -> **Company Y**
2. **Company Y** -> [manufactures] -> **Product Z**
3. **Product Z** -> [uses] -> **Chemical A**

4. **Chemical A** -> [classified as] -> **Environmental Pollutant**

This chain of reasoning, impossible for a standard vector store, enables the system to surface nontrivial insights essential for a thorough analysis.

In next part we will lay foundation as how to design knowledge graph, and subsequent to that part we will code RAG on KG, its exciting !

Artificial Intelligence | Machine Learning | Knowledge Graph | Llm | Rags

Follow

Written by Aniket Hingane | Day Manager, Night Coder

32 Followers

Passionate about simplifying software concepts and design through concise articles. Making complexity accessible, one short piece at a time.

More from Aniket Hingane | Day Manager, Night Coder

Aniket Hingane | Day Manager, Night Coder

LLM apps : Why you MUST know Semantic Router in 2024 : Part 1

Must know LLM apps decision making layer, don't miss out !

· 6 min read · Jan 23, 2024

--

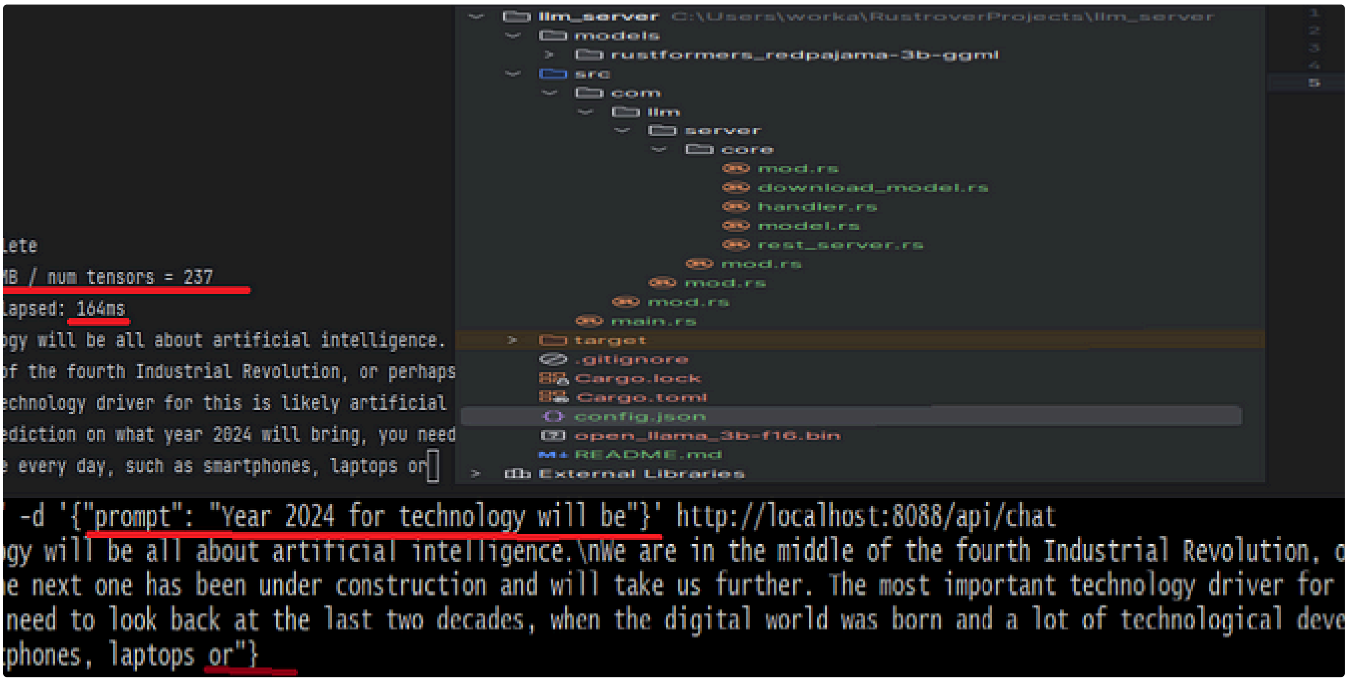
Aniket Hingane | Day Manager, Night Coder

LLM apps : Why you MUST know Semantic Router in 2024 : Part 2

Leveraging Semantic Router for Enhanced Prompting in Your Applications

· 5 min read · Jan 24, 2024

--



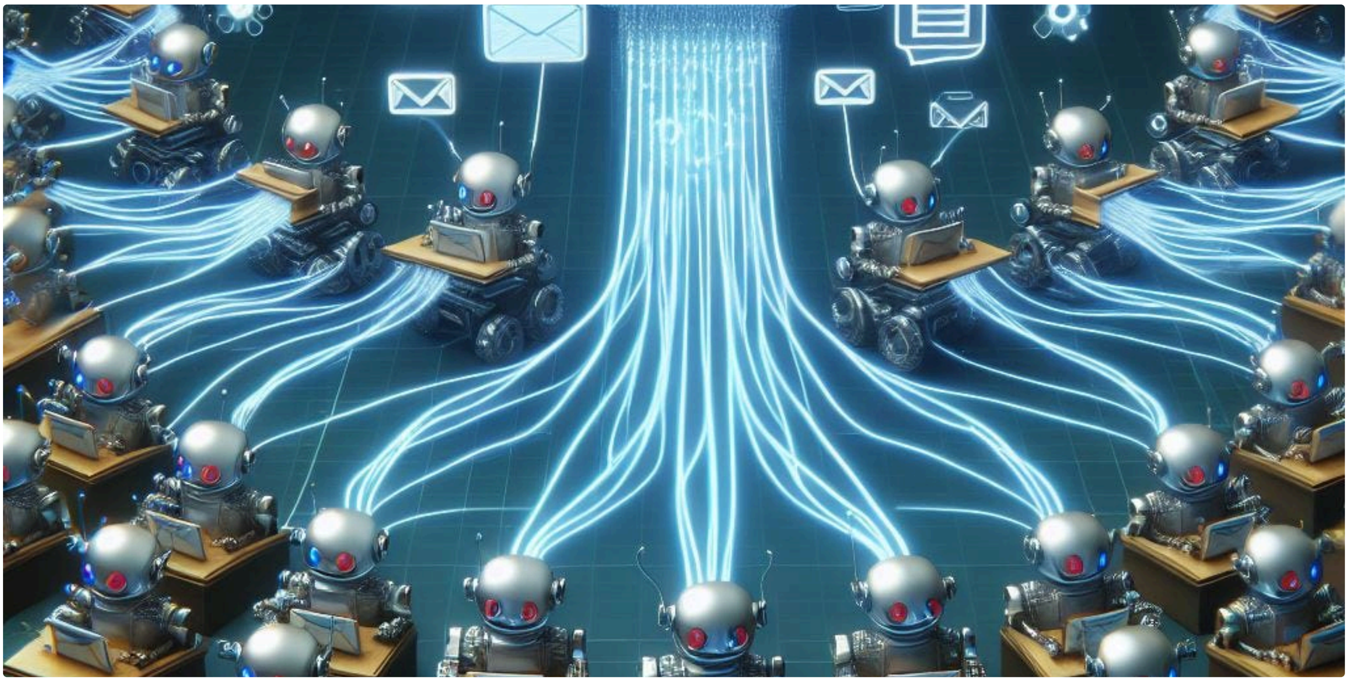
 Aniket Hingane | Day Manager, Night Coder

Must know for AI App Dev : Let’s serve LLMs via Rust programing : LLM Handler Build

Let’s build LLM handler and run super fast inference!

· 5 min read · Feb 18, 2024

--



Aniket Hingane | Day Manager, Night Coder

LLM Apps : Why there is no Prod ready RAG w/o Distributor : 1

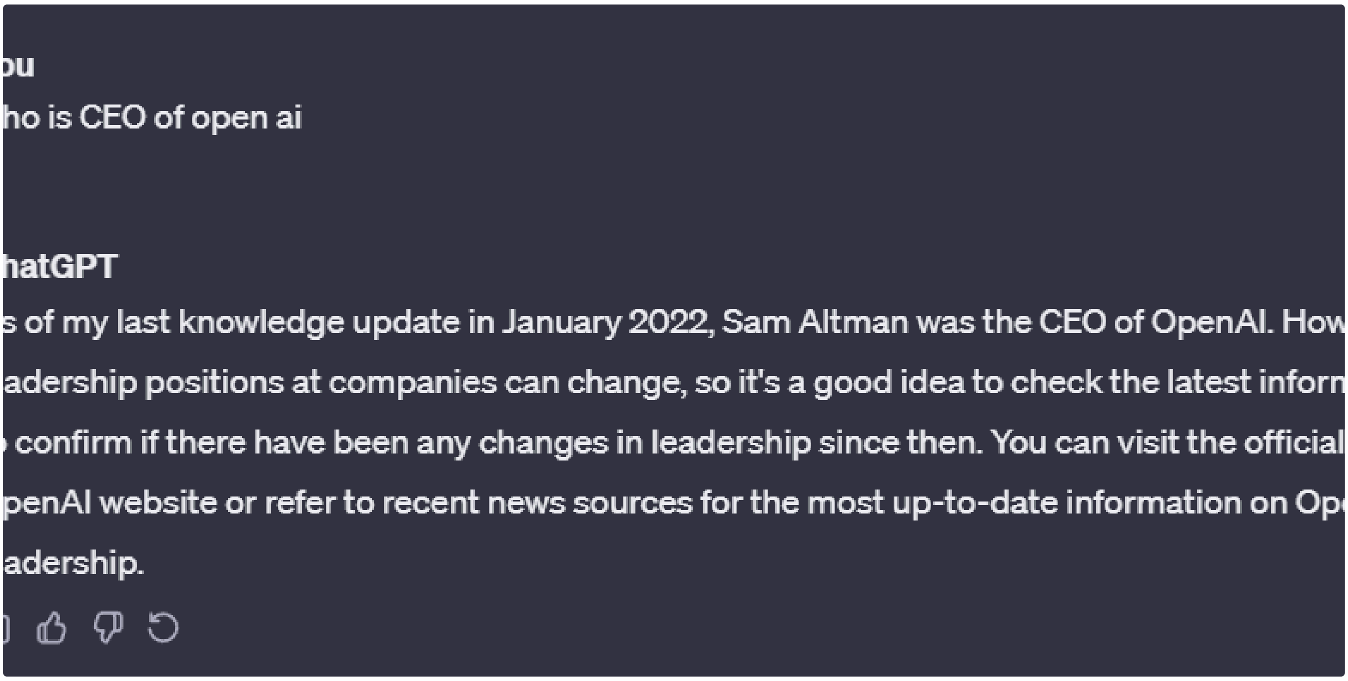
In LLM Apps, why no Prod-ready RAG sans Distributor? Next part we will build it.

· 5 min read · Feb 19, 2024

--

See all from Aniket Hingane | Day Manager, Night Coder

Recommended from Medium



Amit Yadav in FabricHQ

Retrieval Augmented Generation (RAG): Your In-Depth Guide

What is Retrieval Augmented Generation (RAG) and how does it works, let's explore everything in detail.

9 min read · Dec 4, 2023

--

Palantir in Palantir Blog

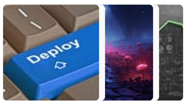
AI's Role in Reimagining the Classification System

The woefully outdated U.S. Government classification system is about to have its AI revolution

5 min read · 6 days ago

--

Lists



Predictive Modeling w/ Python

20 stories · 948 saves

Natural Language Processing

1232 stories · 718 saves

AI Regulation

6 stories · 338 saves

Practical Guides to Machine Learning

10 stories · 1118 saves

Alex Gilmore in Neo4j Developer Blog

PyNeoInstance: A User-Friendly Python Library for Neo4j

Learn how to easily create and query a Neo4j graph using Python. Follow along in the provided demo on GitHub.

6 min read · 5 days ago

--

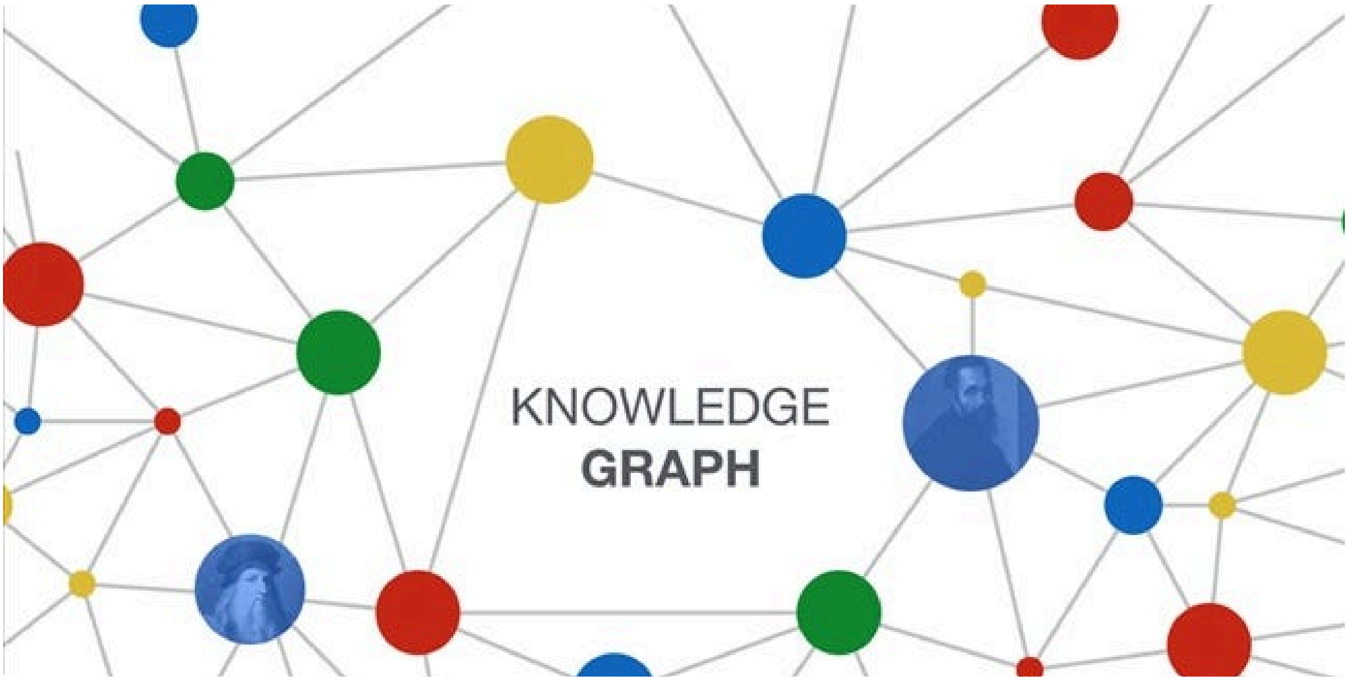
Ninad Kulkarni in Bootcamp

Retrieval-Augmented Generation(RAG) — The secret sauce to supercharge LLMs?

Introduction

8 min read · Dec 24, 2023

--



Phil

Knowledge Graphs in RAGs (with Llama-Index)

Are #KnowledgeGraphs (KGs) better than #VectorDBs in RAGs? Yes and No!

3 min read · Jan 29, 2024

--

Aniket Hingane | Day Manager, Night Coder

LLM Apps : Why Knowledge Graphs are super critical to know if you care about RAG : Let's capture...

Time to explore stackoverflow in different way ! KG way !

· 9 min read · 3 days ago

--

See more recommendations