Question Answering of Knowledge Graph Using Large Language Model

Ritwik Ganguly

Indraprastha Institute of Information Technology

Presentation Overview

Knowledge Graph (KG)

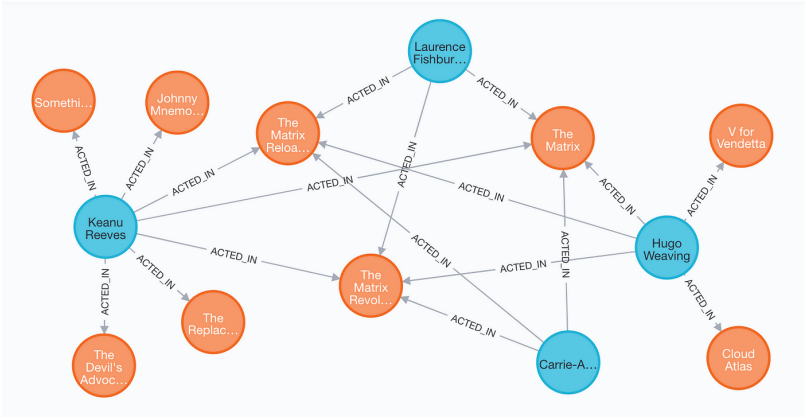
What is KG ?

What is KG ?

Knowledge graphs represent information in an interconnected network of entities and relationships, enabling more complex reasoning across content.

- **Explicit Facts**: Nodes and edges capture facts directly, preserving details.
- **Contextual Details**: Entities include descriptions, aliases, and metadata for context.
- **Network Structure**: Relationships model real-world connections, rules, and timelines.
- **Multi-Hop Reasoning**: Queries traverse relationships to infer facts across sources.
- **Joint Reasoning**: Links unify references to the same object for collective analysis.
- **Explainable Relevance**: Graph connections clarify why facts are relevant.
- **Personalization**: Tailored results based on user attributes and history.

The KG itself a graph data collection, so we need some storage - Neo4j, Mongo DB, Amazon Neptune, Microsoft Azure Cosmos DB, Nebula Graph
I am using here **Neo4j** database, Where the KG is looking like -



Large Language Model

LLM

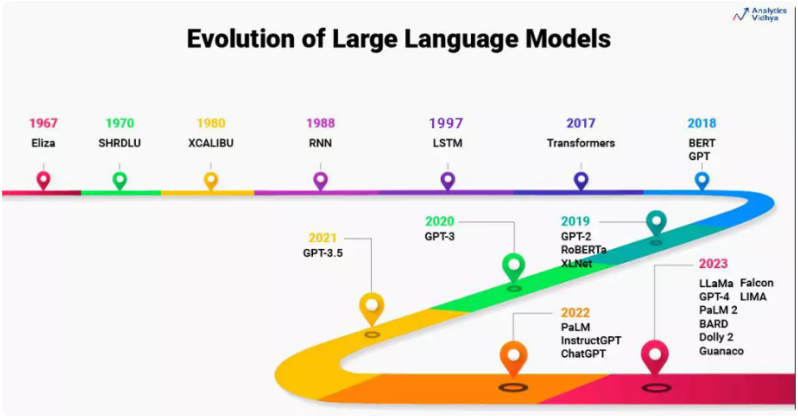
LLM

A type of AI trained on massive text data to understand and generate human-like language.

Main insights:

- **Contextual Understanding:** Analyze and generate text based on the surrounding context.
- **Multilingual Support:** Handle multiple languages fluently.
- **Versatility:** Perform diverse tasks like summarization, Q&A, translation, and content creation.
- **Generative Capability:** Produce human-like, coherent, and context-aware text.
- **Few-shot Learning:** Adapt to new tasks with minimal examples.
- **Scalability:** Leverage large datasets to continuously improve.
- **Interactive Engagement:** Enable dynamic and meaningful conversations.

We are now at the era of AI. where from the wake up to sleep sometimes or many times we pass by the LLM. The evaluation of LLM -



KG with LLM

- Retrieve Augmented Generation (RAG)
- Flowchart
- Methodology
- Prompting

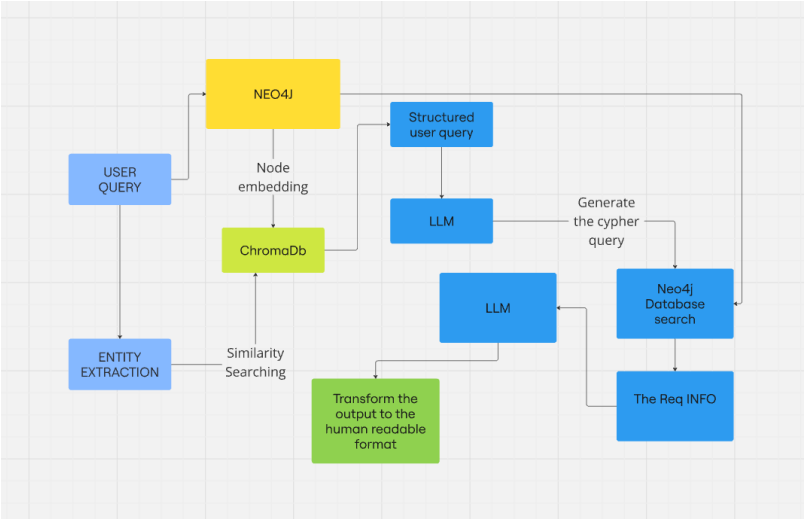
It's a hybrid approach combining the capabilities of retrieval-based systems and LLM to generate contextually relevant and factual responses.
It has two components,

- **Retrieval Component:** Extracts specific and relevant information from a connected database, i.e. KG.
- **Generation Component:** Uses an LLM to analyze and synthesize the retrieved data into coherent and contextually appropriate responses.

Adv of using RAG

- **Mitigates Hallucination:** Grounds LLM responses in factual, retrieved information, reducing incorrect answers.
- **Dynamic Knowledge Integration:** Enables access to up-to-date, domain-specific data from external sources.
- **Scalability:** Efficiently handles large-scale data without retraining LLMs.
- **Explainability:** Provides traceable sources, improving trust and transparency.
- **Personalization:** Tailors responses based on user history and preferences.
- **Cost Efficiency:** Reduces computational costs by targeting retrieval over fine-tuning.
- **Specialized Domains:** Integrates domain-specific data, enhancing versatility in fields like healthcare, law, and finance.

The main methodology of Q&A KG using LLM



Entity Extraction: We need to identify the the main entity on which we need to work.
Through the LLM entity extraction, we can get the main entities.
Node Embedding: It is a technique used to represent the nodes in a KG as low-dimensional vectors. Each vector encodes the structural and semantic properties of the node, such as its relationships with other nodes and its features.

Similarity Searching: Similarity searching using node embeddings involves finding nodes in a knowledge graph (KG) that are most similar to a given node or query.

Each node is represented as a low-dimensional vector (embedding). Similarity is measured by comparing these embeddings using metrics like cosine similarity, Euclidean distance.

Prompting

Prompting To handle a robust and large database, prompting is most important. Prompting in LLM refers to crafting specific inputs to guide the behavior and responses of a LLM. A good prompt ensures clarity, precision, and relevance in the model's output.

- **Be Clear and Specific:** Use unambiguous language to avoid misinterpretation.
- **Provide Context:** Offer sufficient background to guide the model's understanding.
- **Define the Format:** Specify the desired output type (e.g., list, paragraph, table).
- **Avoid Overloading:** Keep prompts concise; avoid unnecessary details.
- **Use Examples:** Illustrate the expected response format when needed.
- **few-shot:** give some understanding examples to tell me llm the direction of thinking.
- **Ask Explicitly:** Directly state the task or question to the model.

Important Docs

Important Docs

1. KG + RAG - Medium TDS
2. Rag system using KG + LLM - KG using RAG
3. Learn KG - Neo4j Official Doc