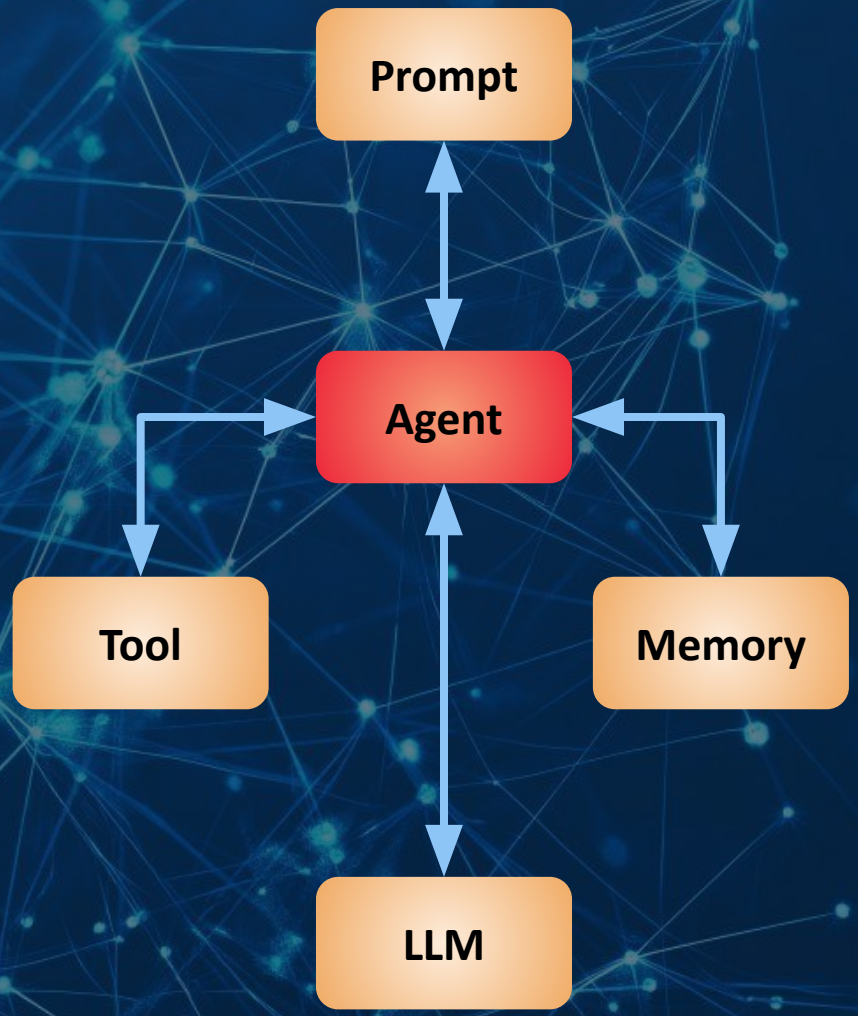
Bridging Language Models with Graph-Driven Optimization

LangGraph

Empowering LLM Agents

What is Agent?



- Utilizes LLM and exhibit capabilities beyond text generation, including
  - Conducting conversations,
  - Completing tasks, reasoning,
  - And can demonstrate some degree of autonomous behaviour.

Agent

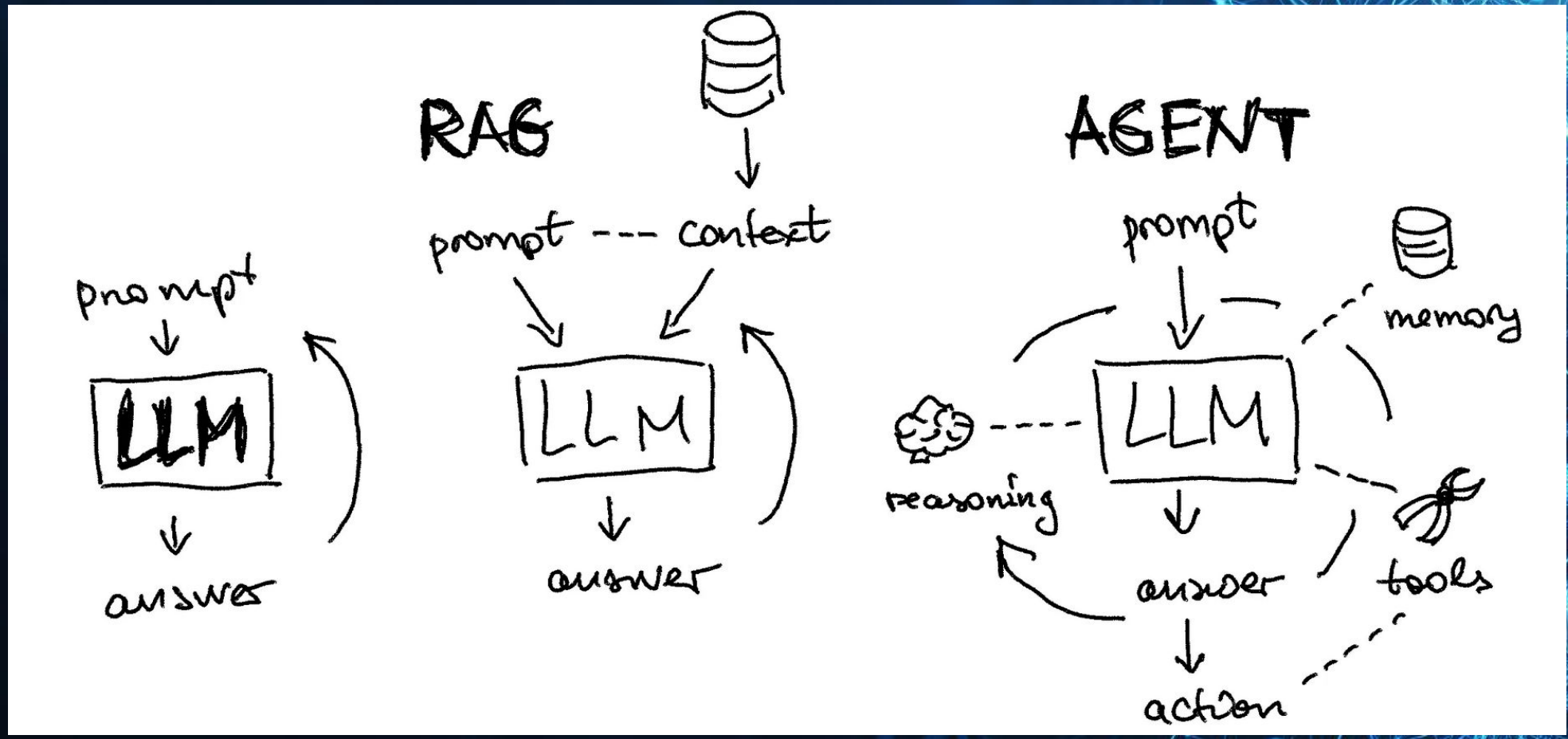
Agents utilize a language model as a reasoning engine to decide the actions to take in which order

- Planning :
  - breaking tasks into smaller sub-goals
- Memory :
  - short term(chat history) / long term(vectorstore)
- Tool Use :
  - It can make use of different tools to extend its capabilities.

What is Tools?

- Agent can interacts with various “Tools” to perform tasks or answer queries.
- Tools are essentially functions that extend the agent's capabilities by allowing it to perform specific actions, like retrieving the current time or accessing an external database.

Agents let the model use tools in a loop, so that it can decide how many times to use tools.



Benefits of AI Agents

- Enhanced Efficiency and Productivity
- Improved Decision-Making
- 24/7 Availability and Scalability
- Personalized Customer Experiences
- Cost Reduction and Increased Revenue
- Innovation and New Opportunities

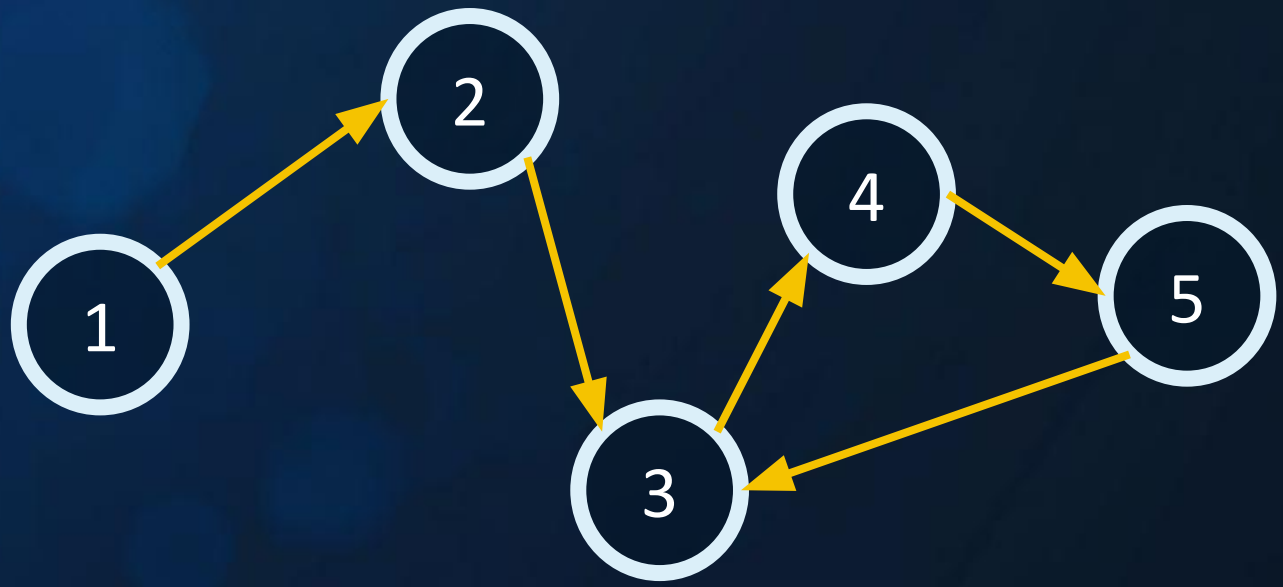
Introduction of Graph

Graph

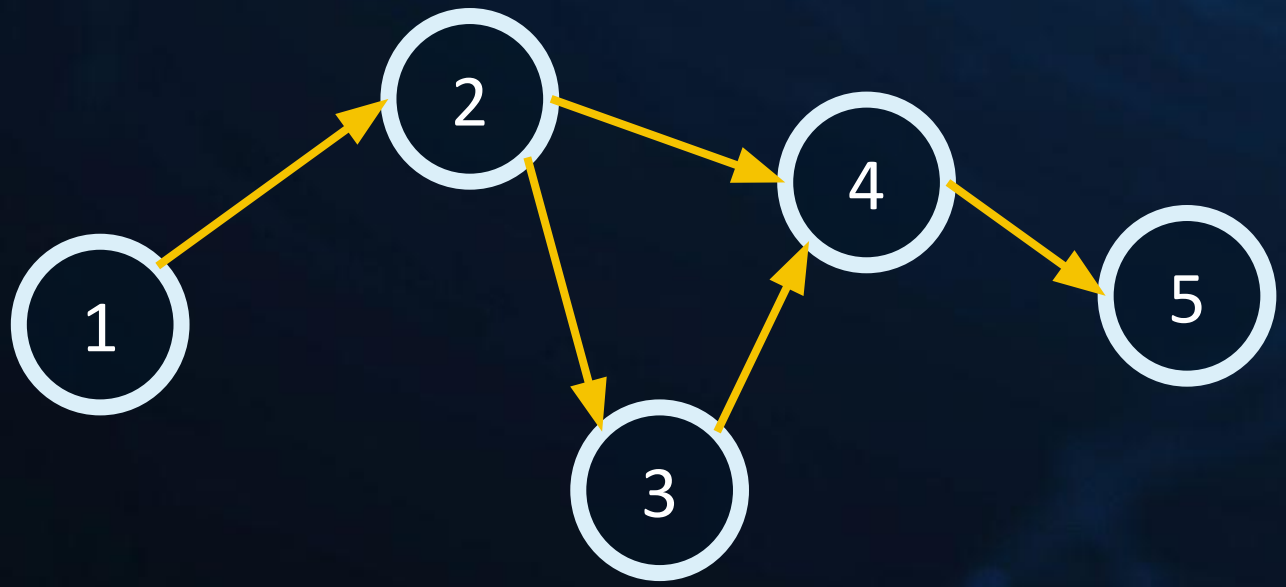
Graph is a collection of edges and Nodes

- Nodes (or vertices): Represent the objects or entities.
- Edges (or links): Represent the connections or relationships between the nodes.
- Directed: Edges have a direction (like an arrow), indicating a one-way relationship.

Directed Cyclic Graph (DCG)



Directed Acyclic Graph (DAG)



folio3 GenAI SIG

Langraph

An introduction to Langraph framework

What is Langchain ?

LangGraph allows you to:

- Create structured pipelines of tasks.
- Visualize and manage complex chains of operations.
- Facilitate conditional execution, parallelization, and branching based on the output of different steps.

What is LangGraph ?

- LangGraph helps manage large, complex workflows where a simple sequential chain of events is insufficient.
- Providing cyclic computational capabilities for LLM applications. While LangChain supports defining computation chains (Directed Acyclic Graphs or DAGs), LangGraph enables the incorporation of cycles.
- This allows for more intricate, Agent-like behaviors, where an LLM can be called in a loop to determine the next action to take.

Key Features of LangGraph

- Graph-Based Workflow
- Parallel Processing
- Conditional Logic
- Error Handling
- Integration with LangChain

Use Case Examples

Data Enrichment Pipeline:

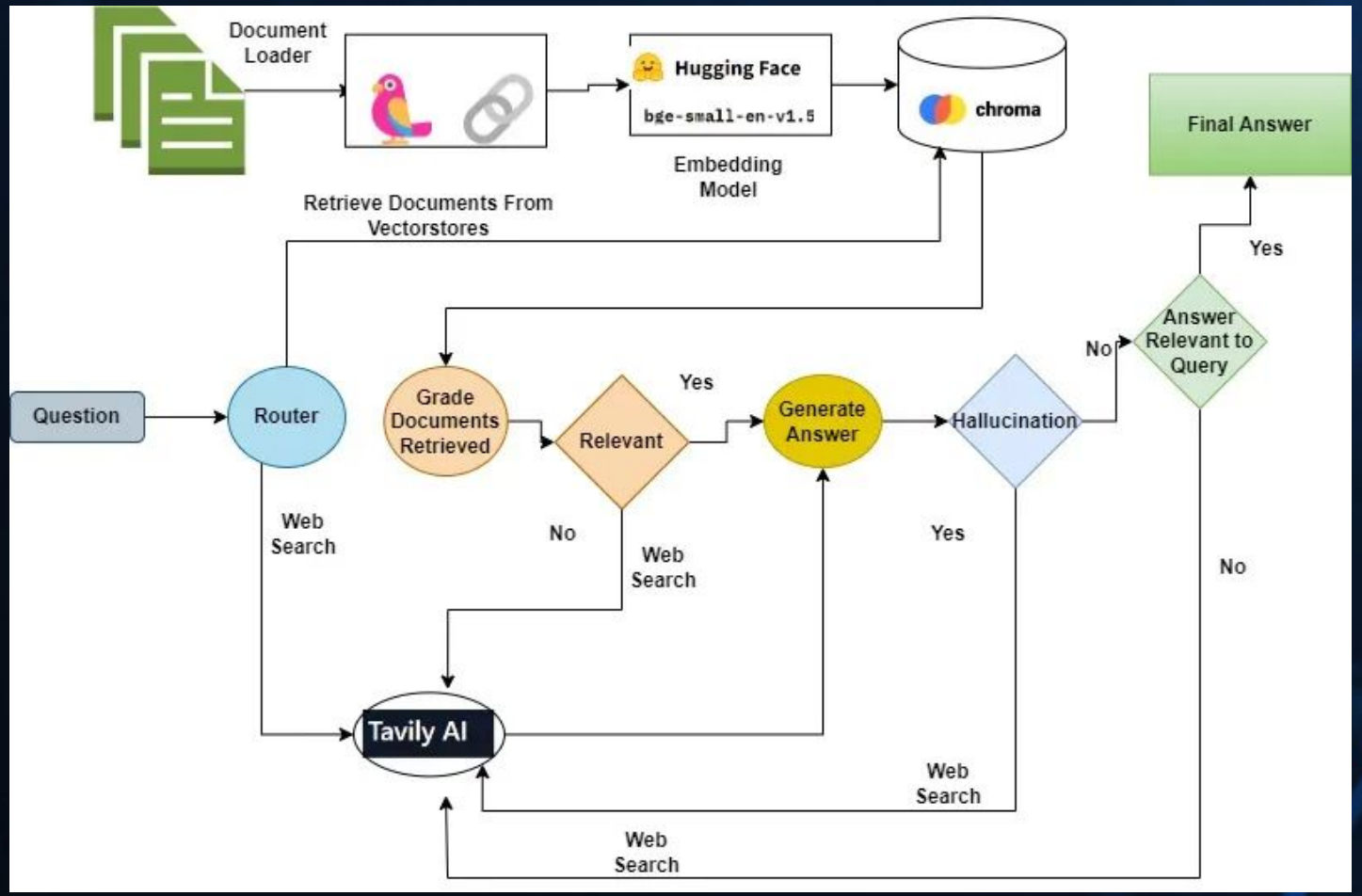
- Step 1: Fetch data from an API.
- Step 2: Run a summarization LLM on the data.
- Step 3: Perform sentiment analysis.
- Step 4: Store the results in a database.

Use Case Examples

Document Analysis:

- Use parallel branches to:
  - Extract named entities.
  - Summarize content.
  - Generate questions from the text.
- Combine the results and return a complete analysis.

Use Case Examples



Creating a Simple Node flow

-

Thank You!