What is LangGraph?

LangGraph overview

LangGraph, created by LangChain, is an open source AI agent framework designed to build, deploy and manage complex generative AI agent workflows. It provides a set of tools and libraries that enable users to create, run and optimize large language models (LLMs) in a scalable and efficient manner. At its core, LangGraph uses the power of graph-based architectures to model and manage the intricate relationships between various components of an AI agent workflow.

What does all this information mean? The following example can offer a clearer understanding of LangGraph: Think about these graph-based architectures as a powerful configurable map, a "Super-Map." Users can envision the AI workflow as being "The Navigator" of this "Super-Map." Finally, in this example, the user is "The Cartographer." In this sense, the navigator charts out the optimal routes between points on the "Super-Map," all of which are created by "The Cartographer."

To recap, optimal routes within the graph-based architectures ("Super-Map") are charted and explored by using the AI workflow ("The Navigator"). This analogy is a great place to start understanding LangGraph—and if you like maps then you are welcome for the bonus opportunity to see someone use the word cartographer.

LangGraph illuminates the processes within an AI workflow, allowing full transparency of the agent's state. Within LangGraph, the "state" feature serves as a memory bank that records and tracks all the valuable information processed by the AI system. It's similar to a digital notebook where the system captures and updates data as it moves through various stages of a workflow or graph analysis.

For example, if you were running agents to monitor the weather, this feature could track the number of times it snowed and make suggestions based on changing snowfall trends. This observability of how the system works to complete complex tasks is useful for beginners to understand more about state management. State management is helpful when it comes to debugging as it allows the application's state to be centralized, thus often shortening the overall process.

This approach allows for more effective decision-making, improved scalability and enhanced overall performance. It also allows for more engagement with individuals who might be new to these processes or prefer a clearer picture of what is going on behind the scenes.

LangGraph is also built on several key technologies, including LangChain, a Python framework for building AI applications. LangChain includes a library for building and managing LLMs. LangGraph also uses the human-in-the-loop approach. By combining these technologies with a set of APIs and tools, LangGraph provides users with a versatile platform for developing AI solutions and workflows including chatbots, state graphs and other agent-based systems.

Delve deeper into the world of LangGraph by exploring its key features, benefits and use cases. By the end of this article, you will have the knowledge and resources to take the next steps with LangGraph.

Key components of LangGraph

Let's begin by first understanding the key components that make up LangGraph. The framework is built around several key components that work together to enable users to create and manage complex

AI workflows. These components include:

Monitoring mechanism

Human-in-the-loop: Human-in-the-loop (HITL) refers to the requirement of human interaction at some point in the process. In the realm of machine learning (ML), HITL refers to a collaborative process where humans augment the computational capabilities of machines to make informed decisions while building a model. By using the most critical data points, HITL enhances the accuracy of machine learning algorithms, surpassing random sampling methods.

Graph architecture

Stateful graphs: A concept where each node in the graph represents a step in the computation, essentially devising a state graph. This stateful approach allows the graph to retain information about the previous steps, enabling continuous and contextual processing of information as the computation unfolds. Users can manage all LangGraph's stateful graphs with its APIs.

Cyclical graph: A cyclical graph is any graph that contains at least one cycle and is essential for agent runtimes. This means that there exists a path that starts and ends at the same node, forming a loop within the graph. Complex workflows often involve cyclic dependencies, where the outcome of one step depends on previous steps in the loop.

Nodes: In LangGraph, nodes represent individual components or agents within an AI workflow. Nodes can be thought of as "actors" that interact with each other in a specific way. For example, to add nodes for tool calling, one can use the ToolNode. Another example, the next node, refers to the node that will be executed following the current one.

Edges: Edges are a function within Python that determines which node to execute next based on the current state. Edges can be conditional branches or fixed transitions.

Tools

RAG: Retrieval-augmented generation (RAG) combines the power of LLMs with contextual information from external sources by retrieving relevant documents, which are then used as input for answer generation.

Workflows: Workflows are the sequences of node interactions that define an AI workflow. By arranging nodes into a workflow, users can create more complex and dynamic workflows that use the strengths of individual components.

APIs: LangGraph provides a set of APIs that enable users to interact with its components in a programmatic way. Users can use an API key, add new nodes, modify existing workflows and retrieve data from an AI workflow.

LangSmith: LangSmith is a specialized API for building and managing LLMs within LangGraph. It provides tools for initializing LLMs, adding conditional edges and optimizing performance. By combining these components in innovative ways, users can build more sophisticated AI workflows that use the strengths of individual components.

Industry newsletter

The latest AI trends, brought to you by experts

Get curated insights on the most important—and intriguing—AI news. Subscribe to our weekly Think newsletter. See the IBM Privacy Statement.
We use your email to validate you are who you say you are, to create your IBMid, and to contact you for account related matters.
Business email

Your subscription will be delivered in English. You will find an unsubscribe link in every newsletter. You can manage your subscriptions or unsubscribe here. Refer to our IBM Privacy Statement for more information.

Your subscription will be delivered in English. You will find an unsubscribe link in every newsletter. You can manage your subscriptions or unsubscribe here. Refer to our IBM Privacy Statement for more information.