5 AI Agent Frameworks Compared

Check out this comparison of 5 AI frameworks to determine which you should choose.

Cornellius Yudha Wijaya

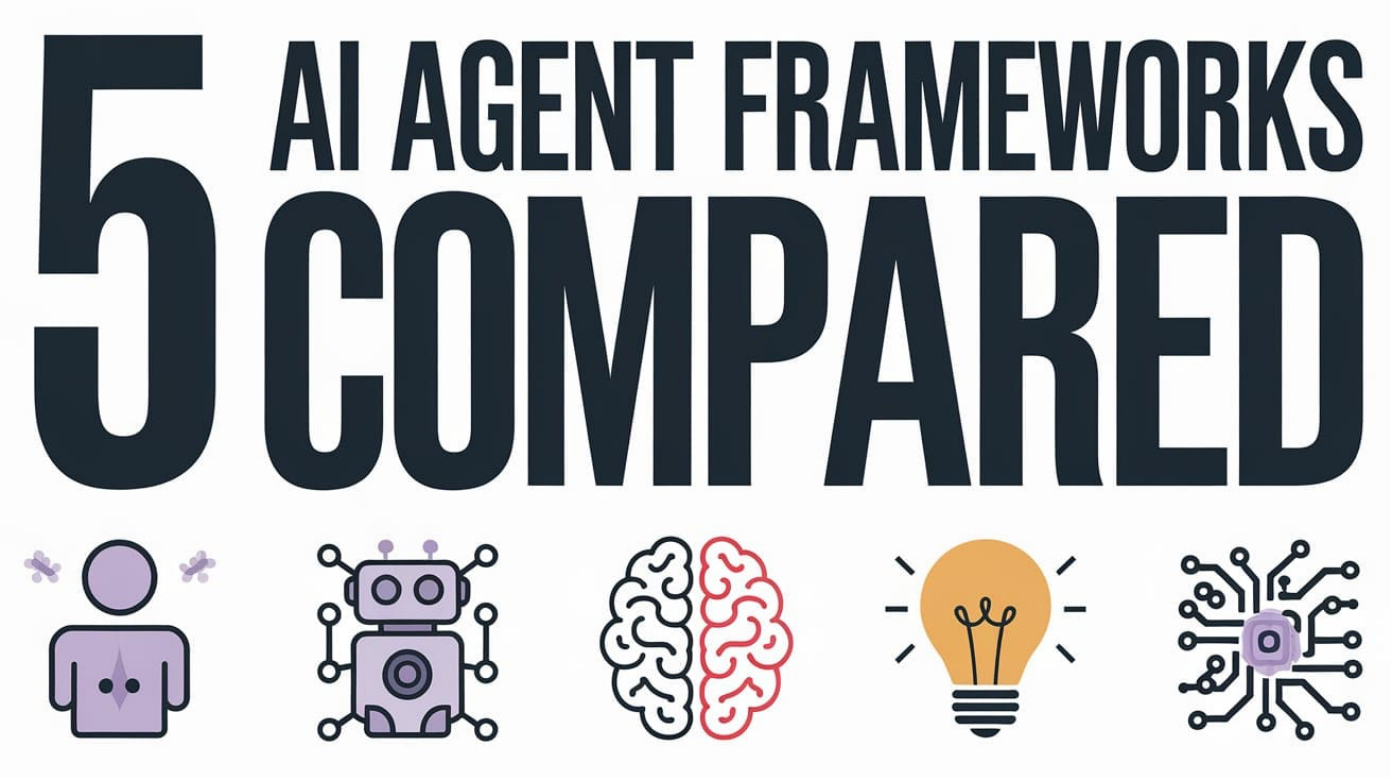

Image by Author | Ideogram.ai

Recently, the topic of AI Agents has been creating a buzz everywhere, initiating a surge of interest as people rush to learn more about them. The interest in the AI Agents topic comes from the increased model development ability in reasoning.

The concept of Agents existed long before LLMs gained popularity. However,

AI Agents have become exponentially more valuable for businesses due to LLMs' enhanced reasoning capabilities, allowing them to perceive their environment and act based on the inputs provided.

With AI Agents' usefulness becoming more apparent, many frameworks for building AI Agents have been developed for people to use. In this article, we will explore five different AI Agents frameworks and compare them in the hope that the reader can select the most useful for your project.

Let's get into it.

LangGraph

The first AI Agent framework we will discuss is LangGraph. As the name states, LangGraph was developed by the developer of LangChain and uses graph-based technology to initiate AI Agent systems. This means we can detail every step and direction the agents take in a way that the graph could be.

The framework is designed to build agents with stateful properties. It offers fine-grained control over the application, focusing on a detailed workflow that involves many complex tasks, such as automated decision-making and multi-step processes. The library is built on top of LangChain, which means it can utilize all the power of LangChain, and in turn, you can use LangSmith for the application lifecycle.

The LangGraph framework is useful when your use cases require dynamic decision-making and human intervention. For example, an agent system where customer support requires human authorization can be beneficial with LangGraph as we can set a time for humans to intervene.

LangGraph is good at offering detailed systems, but it requires a deep

understanding of graph-based workflows and has a steeper learning curve than the other framework. Still, it's a great framework to learn once you have a basic knowledge of AI Agents.

CrewAI

The next AI Agent Framework we will discuss is CrewAI, a Python framework for developing multi-agent systems. The framework facilitates the development of collaboration between agents to solve complex tasks using user-friendly APIs.

The frameworks work by developing AI agents consisting of role-playing agents with defined goals, backstories, and tools. They also assign tasks detailing the goal to be achieved. With all these components, the framework could combine them into collaborative AI systems where agents can focus on their tasks or delegate them by sharing information to achieve a common objective.

The CrewAI framework suits systems requiring agents' teamwork, such as research teams or project management. We can assign each agent a task, such as gathering data, analyzing it, and generating reports. CrewAI can collaborate with agent systems suitable for many tasks by creating a team that focuses on specialized tasks.

CrewAI is relatively more straightforward to use than other frameworks and offers a certain degree of orchestration detail. However, it might present inconsistent results for specific use cases and require higher performance tuning to execute stability in complex tasks.

Smolagents

Smolagents is an AI Agent framework released by the Hugging Face teams.

It is lightweight enough to build any agent and can use many of the resources available in the Hugging Face hub, such as LLM and Tools.

The framework is mainly used for starting agent development and prototyping, focusing on simplicity and speed. You can even quickly build an AI Agent with one-liner code, so it's ideal if you already have a specific idea you want to test.

It performs well for tasks that don't need complex arrangements, such as simple chatbots or answering questions agents. We can even extend the useability and reusability by integrating them with Hugging Face Hub.

As the framework focuses on simplicity, it is unsuitable for large-scale complex interactive agents, especially those requiring a multi-agent system. However, its simplicity might allow them to use other AI Agent frameworks to increase their stability.

Autogen

AutoGen is the next AI Agent framework we will discuss. The Microsoft team developed it to build multi-agent systems using conversational agents. It's one of the earliest frameworks focusing on building AI agentic systems.

The framework supports multi-agent system development designed for scalable and distributed applications, suitable for agents that need to collaborate in real-time environments. It also supports tool execution and function callers, which allow the agent system to perform complex tasks independently.

You can use Autogen when you want to build large-scale applications that need complex systems and real-time data processing, such as financial trading systems or real-time monitoring systems. A lot of use cases that

require real-time aspects will be suitable using Autogen.

It's a complex framework that has a steep learning curve and could result in higher computational resource usage if not executed correctly. That's why, it's not a framework I recommended for your first time building AI Agent.

Phidata

The last AI Agent framework we will discuss is Phidata. It's a multimodal agent framework that can develop agentic systems that perform collaboratively. It's also designed to work with components such as memory and tools to help them perform autonomously and consistently.

By default, Phidata agent support multimodal data such as text, images, and audio data which make them valuable without a need to rely on external tools. The framework also provide Agentic UI to use if you more into visual interactions with your agents. They also one that pioneering Agentic RAG where the agent can search the knowledge base.

Phidata is suitable for developing system that required domain-specific with lot of specialized agents that need to work together. For example, AI Assistant for financial trading or research development.

It's easy to use but Phidata still have steeper learning curve if you want to make the system stable in production. The resources is also higher if it's not setup right, so you need to careful in defining the agents.

Conclusion

In this article, we have discuss and compare five different AI Agent frameworks that we can use to build AI Agentic system. The frameworks that we have discuss are:

- LangGraph: For detailed, stateful systems with human intervention.
- CrewAI: For collaborative, multi-agent teamwork.
- Smolagents: For fast prototyping and lightweight tasks.
- Autogen: For real-time, scalable systems.
- Phidata: For multimodal, domain-specific collaboration.

I hope this has helped!

Cornellius Yudha Wijaya is a data science assistant manager and data writer. While working full-time at Allianz Indonesia, he loves to share Python and data tips via social media and writing media. Cornellius writes on a variety of AI and machine learning topics.

More On This Topic

- Understanding Agent Environment in AI
- Generative Agent Research Papers You Should Read
- Frameworks for Approaching the Machine Learning Process
- Chip Huyen shares frameworks and case studies for implementing ML systems
- Expert Insights on Developing Safe, Secure, and Trustworthy AI Frameworks
- Types of Visualization Frameworks