What is AutoGen?

Microsoft AutoGen is an open-source framework for building AI agents and other artificial intelligence applications. It is a result of Microsoft Research’s foray into agentic AI, simplifying the creation of multi-agent systems using large language models (LLMs).

An award-winning 2024 paper from Microsoft’s Chi Wang and other researchers demonstrated AutoGen’s applicability to several real-world problems, including supply-chain optimization and online decision-making.[1] AutoGen’s Python SDK makes getting started is simple as a pip install command.

While AutoGen is one leading multi-agent framework, there is a whole ecosystem of AI agent frameworks to choose from. Others include crewAI, LangChain and LangGraph, as well as IBM’s BeeAI.

AutoGen's architecture

AutoGen is composed of three main layers.

The core layer

Core is AutoGen’s foundational layer, the basic plumbing and wiring that makes the AutoGen framework function. In Microsoft’s language, “Core API implements message passing, event-driven agents, and local and distributed runtime.” In other words, it allows the agents to talk to each other, empowers them to wake up upon certain event triggers and enables them to run locally on your computer or across various servers.

The AgentChat layer

If Core is plumbing and wiring, AgentChat is something like a prefab home with built-in fixtures. AgentChat presumes (based on prevailing use cases) that most people want AI agents to be able to chat to humans and other bots (in technical terms, to be “conversable agents”). And rather than force developers to code an orchestration logic from scratch, AgentChat further assumes that in multi-agent collaboration, there will be a division of labor, with agent teams frequently including an “AssistantAgent” (which uses LLMs to do “thinking” for the user) as well as a “UserProxyAgent” (for code execution and tool use). This ability to leverage “template” agent teams helps facilitate rapid prototyping of AI applications.

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.

The extensions layer

AutoGen is "extensible," meaning users can add new capabilities. AutoGen's default Extensions includes components like LocalSearchTool, which enables search within your own set of files, as well as MultimodalWebSurfer, which can surf the broader internet. Microsoft encourages developers to create their own extensions as well.

Additional helpful tools include AutoGenBench, which benchmarks agentic AI performance and helps direct debugging, as well as AutoGen Studio, a no-code interface for beginners (for which an approachable video tutorial can be found on YouTube).

Real-world use cases of AutoGen

Microsoft has claimed to see hundreds of applications of AutoGen in industries ranging from biotech to consumer packaged goods to telecommunications.[2]

Education

A professor of physical therapy at Tufts University, Benjamin Stern, has used AutoGen for complex tasks, including the creation of tailored assessments, individualized study guides and tutoring for students transitioning to graduate-level courses. Additionally, he has used agent interactions to simulate patient interviews and has leveraged AutoGen's "group chat"-like capabilities to foster round-robin debate formats. He also reports using OpenAI Assistant agents through AutoGen.

Drug discovery

The pharmaceutical firm Novo Nordisk has reported several ways it uses Microsoft's AI stack to conduct and share reasoning in drug discovery.[3] Sam Khalil, the company's VP of data nsights, reports that AutoGen is "helping us develop a production-ready multi-agents framework."

Data science

IBM engineers Kelly Abuelsaad and Anna Gutowska have created a Multi-agent RAG with AutoGen application that works from human inputs to gather information from a local corpus of documents. They describe a system where six highly specialized agents (including a planner agent, a research assistant and a report generator, among others) divide and conquer. "No longer do we need to write complex SQL queries to extract relevant data from a knowledge base," they write. The solution is more scalable than working with one big model, since developers can selectively augment any single agent that becomes a bottleneck.

Occupational safety

On Github, one user has demonstrated how AutoGen could be used to examine images taken from a camera in a potentially dangerous environment like a factory, determining in real-time if any humans present are not wearing a helmet. Through an automation, the system would add a red bounding-box on top of the image to alert safety personnel.

AutoGen vs. AG2

The above has described AutoGen, the Microsoft offering. However, as is so often the case with software projects, there has been a fork in the road. A competing framework, AG2, is touted as an "Open-Source AgentOS for AI Agents" by its creators, including the aforementioned Chi Wang. Formerly at Microsoft, Chi Wang later moved on to Google DeepMind; he appears to have decided to evolve an independent version of AutoGen since leaving Microsoft.

"This isn't a new framework — it's basically AutoGen 0.2.34 continuing under a new name," according to one Reddit user who sought to alleviate confusion.[4] One of the main differences between Micosoft's AutoGen and AG2 is that the latter is community-driven, rather than backed by one large firm. Maintainers of AG2 include Wang, as well as researchers from Meta, IBM, and various universities.[5]