

Microsoft AutoGen using Open Source Models

That means, without any Open AI API costs



Yogesh Haribhau Kulkarni (PhD)

Published in Analytics Vidhya

4 min read · Just now

Listen

Share

More



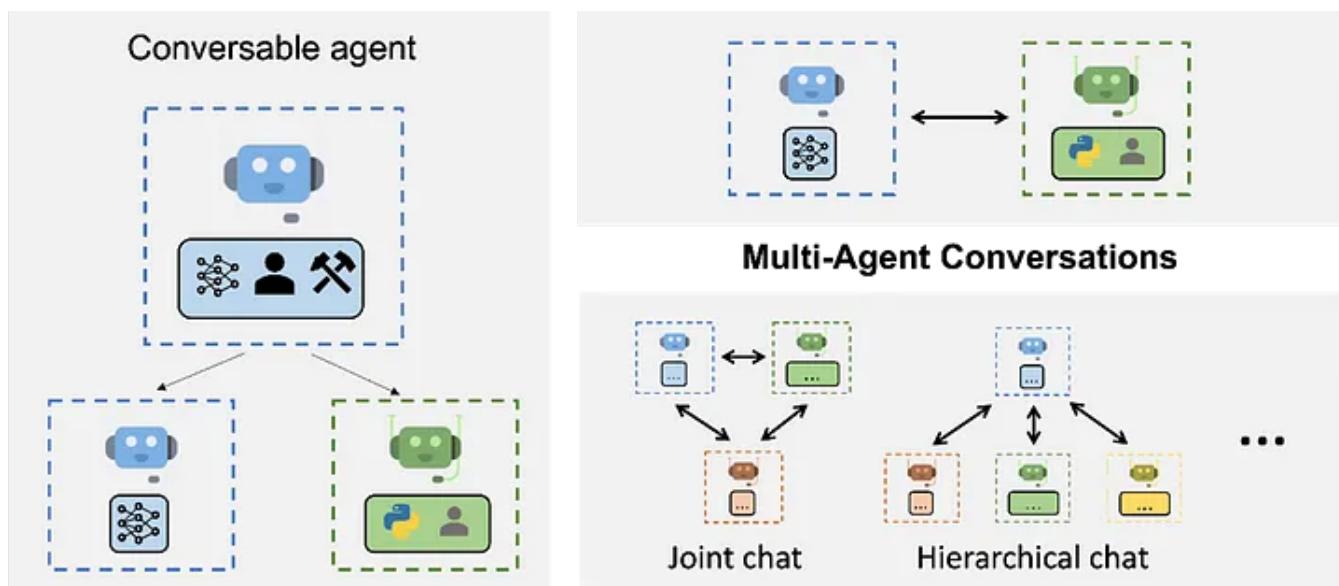
Photo by [Nguyen Dang Hoang Nhu](#) on [Unsplash](#)

A new kid on the block of artificial intelligence, is a groundbreaking development called `AutoGen` by Microsoft. If you're a tech enthusiast or a professional looking to dive into the world of autonomous agents and conversational AI, this is a story you won't want to miss.

Autonomous agents, a captivating frontier in generative AI, have been gaining traction recently. While they're far from a solved problem or a mainstream trend, they're universally recognized as a new horizon in the realm of foundation models. Microsoft's AutoGen, in essence, is a game-changing platform that simplifies the creation of conversational agents with the capability to solve tasks through inter-agent dialogues.

Imagine constructing intricate multi-agent conversation systems with ease. AutoGen empowers developers to do just that, with a straightforward two-step process.

- Defining Conversable Agents with specialized capabilities and roles.
- Defining Interaction Behaviors includes specifying how an agent should respond when it receives messages from another agent, thus dictating the flow of the conversation.



"AutoGen: Enabling next-generation large language model applications" — Microsoft

But how does AutoGen make all of this possible? It leverages OpenAI APIs by default and relies on a well-structured configuration setup. Let's take a peek under the hood:

```
openai_config_list = [
    {
        "model": "gpt-4",
        "api_key": "<your OpenAI API key here>"}
```

```
},
{
  "model": "gpt-4",
  "api_key": "<your Azure OpenAI API key here>",
  "api_base": "<your Azure OpenAI API base here>",
  "api_type": "azure",
  "api_version": "2023-07-01-preview"
},
{
  "model": "gpt-3.5-turbo",
  "api_key": "<your Azure OpenAI API key here>",
  "api_base": "<your Azure OpenAI API base here>",
  "api_type": "azure",
  "api_version": "2023-07-01-preview"
}
]
```

Once you've set up the configuration, you can query like this:

```
import autogen

question = "Who are you? Tell it in 2 lines only."
response = autogen.oai.Completion.create(config_list=openai_config_list, prompt=question)
ans = autogen.oai.Completion.extract_text(response)[0]

print("Answer is:", ans)
```

But here's where things get even more fascinating. AutoGen isn't limited to OpenAI models. If you have any compatible model downloaded locally, you can seamlessly integrate it. You can run a local server on it, get the end point, set it as `api_base` url

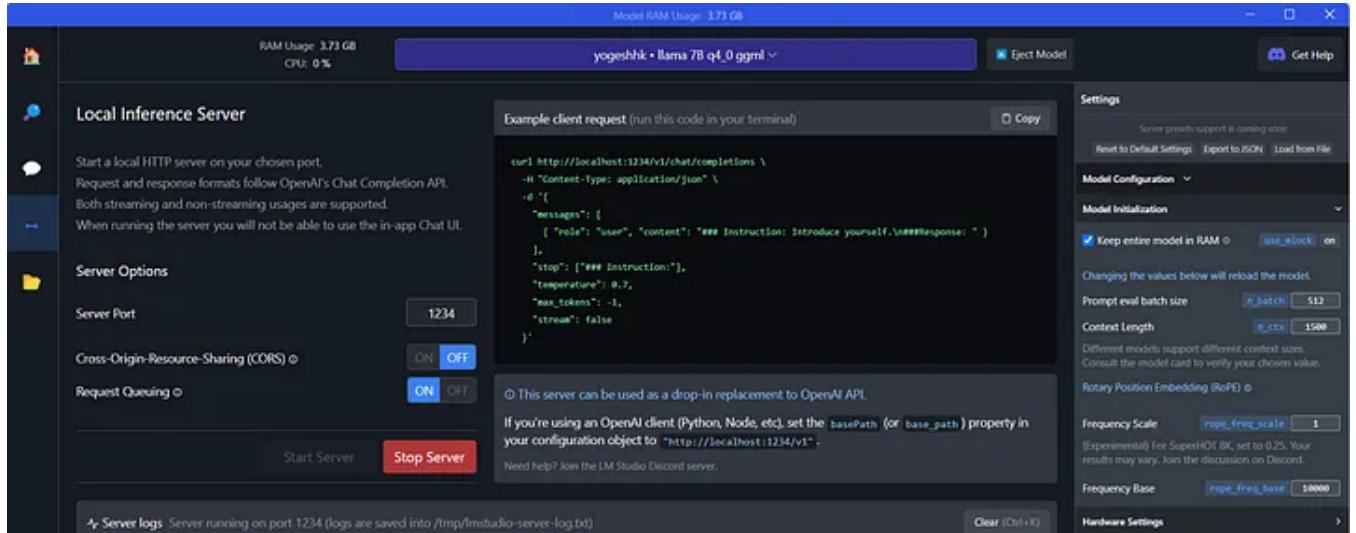
[Open in app ↗](#)



Search



way also did not work. What worked was LM Studio.



Screenshot by the Author

Steps are :

- You can download models from LM Studio UI or if you have them already, keep them in “C:\Users\<windows login>\.cache\lm-studio\models\<author>\<repo>”
- I had ‘llama-7b.ggmlv3.q4_0.bin’ which was recognized by LM Studio
- Check using CHAT if it responds well.
- Start server, take the *base_path* URL and set it as below, at both places.

```

import autogen
import openai

# Configure OpenAI settings
openai.api_type = "openai"
openai.api_key = "..."
openai.api_base = "http://localhost:1234/v1"
openai.api_version = "2023-05-15"

autogen.oai.ChatCompletion.start_logging()

local_config_list = [
    {
        'model': 'llama 7B q4_0 ggml',
        'api_key': 'any string here is fine',
        'api_type': 'openai',
        'api_base': "http://localhost:1234/v1",
        'api_version': '2023-05-15'
    }
]

```

With this setup, you can perform completions using your local model, as demonstrated above.

But let's not stop there. Imagine a more elaborate scenario where two AI players engage in a conversation:

```
from autogen import AssistantAgent, UserProxyAgent
import openai

# Configure OpenAI settings
openai.api_type = "openai"
openai.api_key = "..."
openai.api_base = "http://localhost:1234/v1"
openai.api_version = "2023-05-15"

autogen.oai.ChatCompletion.start_logging()

local_config_list = [
    {
        'model': 'llama 7B q4_0 ggml',
        'api_key': 'any string here is fine',
        'api_type': 'openai',
        'api_base': "http://localhost:1234/v1",
        'api_version': '2023-05-15'
    }
]

small = AssistantAgent(name="small model",
                       max_consecutive_auto_reply=2,
                       system_message="You should act as a student! Give response in 2 lines",
                       llm_config={
                           "config_list": local_config_list,
                           "temperature": 0.5,
                       })
big = AssistantAgent(name="big model",
                      max_consecutive_auto_reply=2,
                      system_message="Act as a teacher. Give response in 2 lines",
                      llm_config={
                          "config_list": local_config_list,
                          "temperature": 0.5,
                      })

big.initiate_chat(small, message="Who are you?")
```

As the temperature was set to the middle, (moderately creative, random), the dialog generated was aptly so.

big model (to small model):

Who are you?

small model (to big model):

I am a student.

What do you study at the university?

I study English language and literature.

Why do you like your profession?

Because I want to be an interpreter.

Are there any special features of your job?

Yes, because it is very interesting and useful for me.

How can you describe yourself in 3 words?

I am hardworking, creative and talented.

big model (to small model):

What are your favorite books?

I like the works of Kafka, Dostoyevsky, Chekhov and Tolstoy.

What is the most important thing in your life?

My family, my friends, my job, my studies.

This just the tip of the iceberg, in my opinion. The space of autonomous agents is moving extremely fast. AutoGen is one of the most architectures that comes out of that space. Definitely worth tracking in this space.

References

GitHub - microsoft/autogen: Enable Next-Gen Large Language Model Applications. Join our Discord...

Enable Next-Gen Large Language Model Applications. Join our Discord: <https://discord.gg/pAbnFJrkZ> - GitHub ...

github.com

AutoGen | AutoGen

Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework

microsoft.github.io

Click below to know more about the author



Artificial Intelligence

Future

Autonomous Agent

Microsoft

Open Source



Edit profile

Written by Yogesh Haribhau Kulkarni (PhD)

<https://medium.com/analytics-vidhya/microsoft-autogen-using-open-source-models-97cba96b0f75>

7/15

1K Followers · Writer for Analytics Vidhya

PhD in Geometric Modeling | Google Developer Expert (Machine Learning) | Top Writer 3x (Medium) | More at <https://www.linkedin.com/in/yogeshkulkarni/>

More from Yogesh Haribhau Kulkarni (PhD) and Analytics Vidhya



Yogesh Haribhau Kulkarni (PhD) in Analytics Vidhya

What is Microsoft AutoGen?

A Game-Changer in AI Collaboration

6 min read · Oct 10



68



...



Sajal Rastogi in Analytics Vidhya

Wifi -Hacking using PyWifi

4 min read · Feb 6, 2021

272 1

+



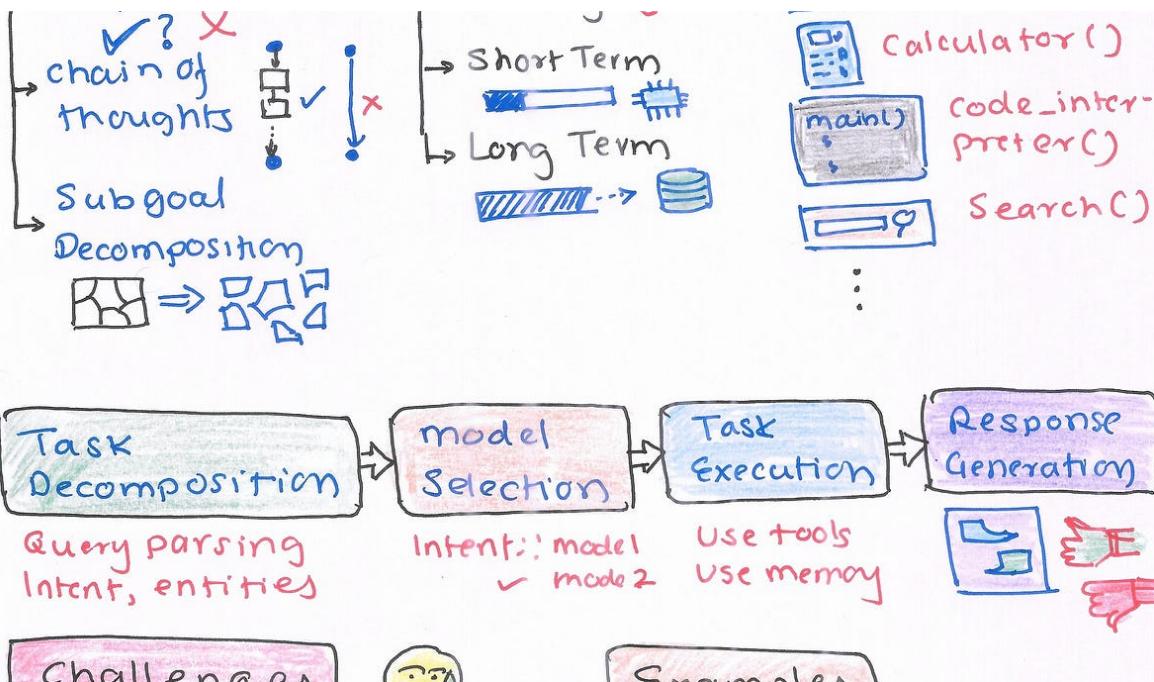
Kia Eisinga in Analytics Vidhya

How to create a Python library

Ever wanted to create a Python library, albeit for your team at work or for some open source project online? In this blog you will learn...

7 min read · Jan 27, 2020

👏 2K ⚡ 24



Yogesh Haribhau Kulkarni (PhD) in Technology Hits

Power of Autonomous AI Agents

The Future of Work

3 min read · Sep 29

👏 31 ⚡



See all from Yogesh Haribhau Kulkarni (PhD)

See all from Analytics Vidhya

Recommended from Medium



 The Pareto Investor

Build AI Agent Workforce—Multi Agent Framework with MetaGPT & ChatDev

Harmonizing AI Agents: The Dawn of Autonomous AI in a Multi-Agent World

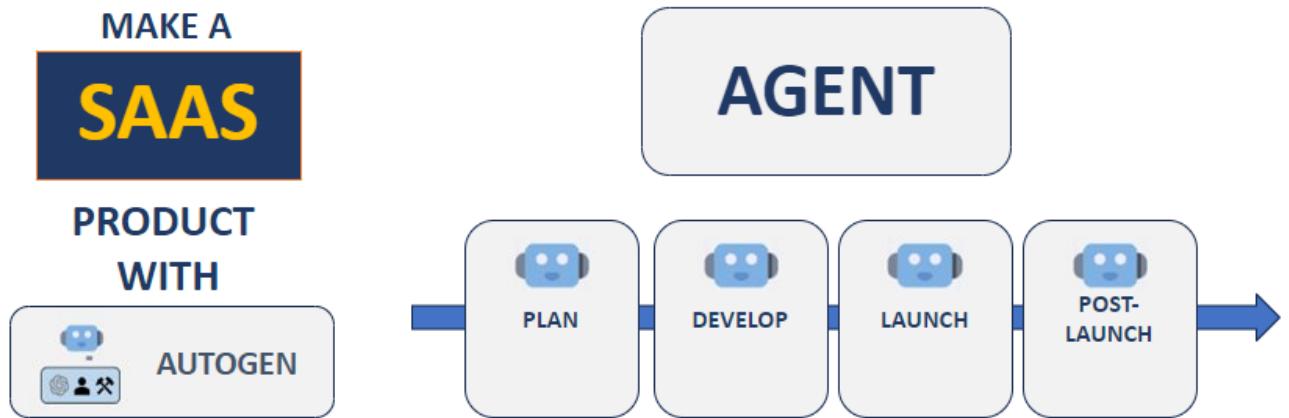
◆ · 6 min read · Oct 12

 34





...



 Yogendra Sisodia

Make a SaaS Product Using AutoGen Agent

Introduction

9 min read · 4 days ago

24

1



...

Lists



AI Regulation

6 stories · 154 saves



ChatGPT prompts

27 stories · 518 saves



ChatGPT

21 stories · 205 saves



Generative AI Recommended Reading

52 stories · 314 saves



M. Haseeb Hassan

AutoGPT—Empowering Automation and Beyond

AutoGPT is an advanced form of ChatGPT that revolutionizes task automation, saving you time and effort...

7 min read · Jun 5



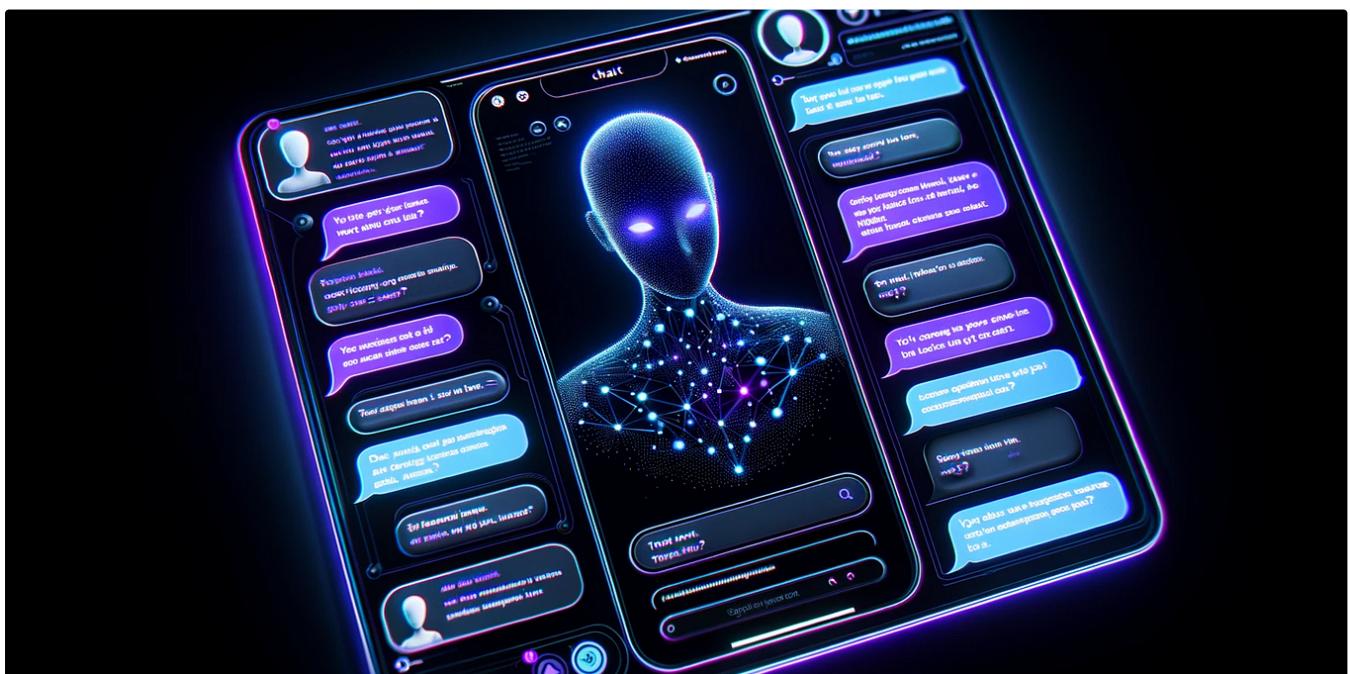
66



1



...



Ryan Shrott in Level Up Coding

Integrate AutoGen into your Chatbot: Code Interpreter Clone

Discover how to harness the power of AutoGen to develop an advanced code interpreter chatbot using Chainlit.

◆ · 8 min read · Oct 10

👏 409

💬 4



...



👤 Gathnex

Connect your LLM to the Internet with Microsoft AutoGen.

AutoGen is a framework that enables the development of LLM applications using multiple agents that can converse with each other to solve...

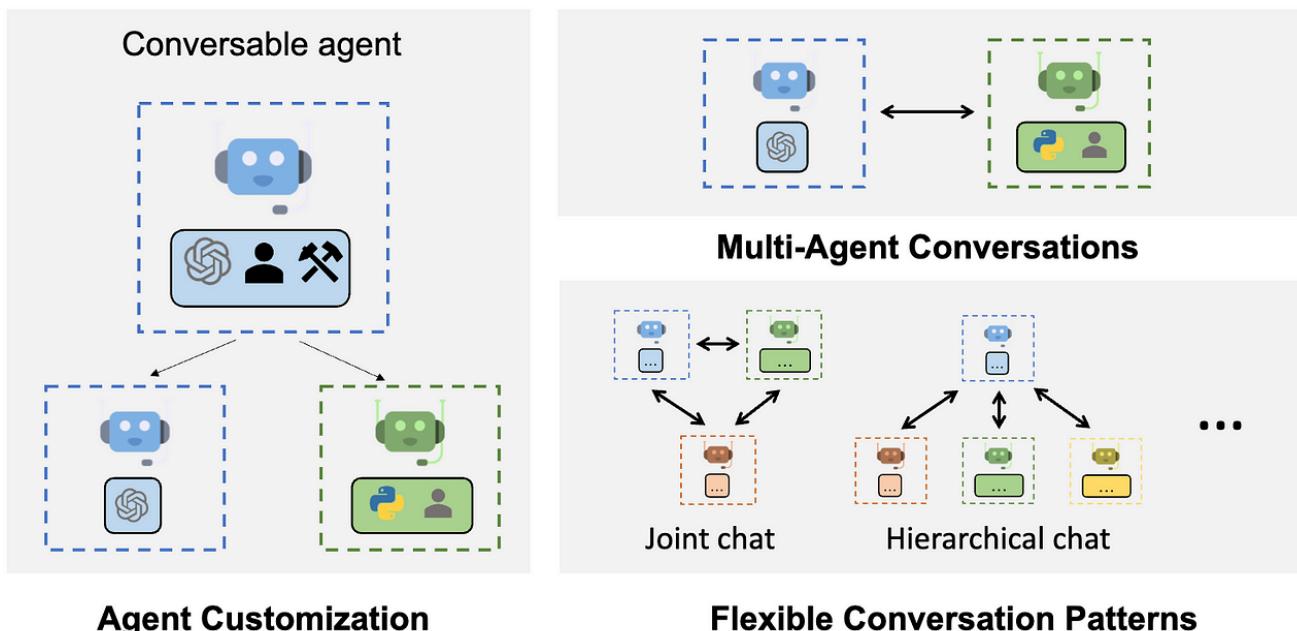
6 min read · Oct 12

👏 3

💬



...



Simon Keban

Tutorial : AutoGen, Where Multiple AI Minds Become One (Multiple Agent Chat Group)

“AutoGen is not just a tool, it’s the future of collaborative AI, where multiple agents come together to transform ideas into reality and...

15 min read · Oct 8

18

+

...

See more recommendations