Two minds – how to multiply ChatGPT →GitHub repository

Sławomir Wiśniewski MD PhD

SCI4BIZ

Introduction

Today's world of AI technology is evolving at an incredible pace, and the capabilities it offers are advancing. One of the key developments is the ability to effectively collaborate between different AI models in a single environment. In this project, we present an innovative approach to integrating two GPT accounts into a single communication platform - Slack.

Why two GPT instances? Two GPT accounts running simultaneously offer unique capabilities. They can work in a complementary way, analyzing different aspects of the same problem, cooperating in solving complex tasks or even controlling and verifying the responses of the other model. Integration with the Slack platform, on the other hand, makes it easy and intuitive to communicate with the models and use their abilities in real, everyday communication situations. Our project aims not only to present the technical aspects of such integration, but also to explore the practical benefits and challenges of collaboration between two AI models. We invite you to learn more about our solution and join the discussion on the future of collaboration in the AI world.

If you would like to support this project, please consider donating.

Your support will help to further develop and sustain this project.

You can donate with STRIPE or Viva Wallet

Scan to donate

Scan to donate

Pre-Req

To effectively use the solution on a desktop computer and enable Slack and two GPT accounts, the user should meet the following prerequisites:

Hardware:

1. Processor: Recommended at least a mid-range multi-core processor (e.g. Intel Core i5 or better).
2. RAM: Minimum 8 GB of RAM, recommended 16 GB or more for smooth operation.
3. Disk space: Recommended at least 10 GB of free space on a hard drive or SSD.
4. Internet connection: Stable and fast internet connection, as communication with the OpenAI API and Slack takes place online.

Software:

1. Operating system: Recommended systems are Windows, macOS or Linux.
2. Python: Python version 3.6 or later installed. Many libraries and scripts in the solution will require Python to run.
3. Python libraries: Installed appropriate Python libraries required for the scripts to run (e.g. slack-sdk, openai, etc.). These can be installed using the pip tool.
4. Developer tools: It is recommended to have an integrated development environment (IDE) such as Visual Studio Code, PyCharm or similar to easily manage code and debug any problems.

Accounts and API keys:

1. Slack account: The user must have a Slack account and permissions to create applications/bots on the selected channel or workspace.
2. OpenAI API Keys: To use ChatGPT, the user must have API keys from OpenAI. These keys are needed for authorization and communication with the ChatGPT service.

Knowledge and skills:

1. Basic knowledge of Python: To configure and customize the scripts, the user should be familiar with the basics of the Python language.
2. Understanding of APIs: A basic understanding of how APIs work, particularly the Slack and OpenAI APIs, will be helpful in managing and troubleshooting integration issues.

Security:

Security: It is recommended to use secured connections (HTTPS) and store API keys in a secure location (e.g., using password managers or environment variables).

Integration steps

Directly connecting two GPT-4 instances for collaborative project work is not a standard feature provided by OpenAI. However, there are methods that can be employed to simulate collaboration between two models:

I. Create an account on Slack, create an application and obtain the necessary tokens

1. create an account on Slack and create a workspace
2. go to https://api.slack.com
3. click "Create an app" and select "From scratch"
4. give the app a name such as BotA (for the next BotB) and select your created workspace
5. this point differs for both bots.

BotA	BotB
5. under "Basic Information" > "Add features and functionality", click on "Permissions". Under "Scopes", add the following permissions for Bot Token Scopes: app_mentions:read channels:history channels:read chat:write 6. in the settings, click on "Socket Mode", enable it and give the token a name. Copy the Slack Bot app token (it starts with "xapp"). 7. under "Basic Information" > "Add features and functionality", click on "Event Subscriptions" and enable it. Then in the "Subscribe to bot events" section, select "app_mention". Save the changes.	5. under "Basic Information" > "Add features and functionality", click on "Permissions". In the "Scopes" tab, add the following permissions for Bot Token Scopes: channels:history channels:read chat:write 6. in the settings, click on "Socket Mode", enable it and give the token a name. Copy the Slack Bot application token (it starts with "xapp"). 7. under "Basic Information" > "Add features and functionality", click on "Event Subscriptions" and enable it. Then in the "Subscribe to bot events" section, select "message.channels" (to respond to messages in the channel). Save the changes.

8. Go to the ”OAuth&Permissions” section and install your app to your workspace.

9. copy the Slack Bot Tokens (starts with “xoxb”)

II. Get OpenAI API key

1. go to the OpenAI API page and log in (create an account if you don't have one)

2. go to the API Key section

3. create a new API key and copy it (save and store it in a safe place - this is sensitive data)

4.repeat with second OpenAI account

III. Install necessary dependencies

pip install openai slack-bolt slack

IV. In the scripts from the Github repository, insert the tokens and OpenAI key obtained in the previous points (I. and II). Be thorough. Then run both Bots and test their capabilities.

V. ENJOY!!!