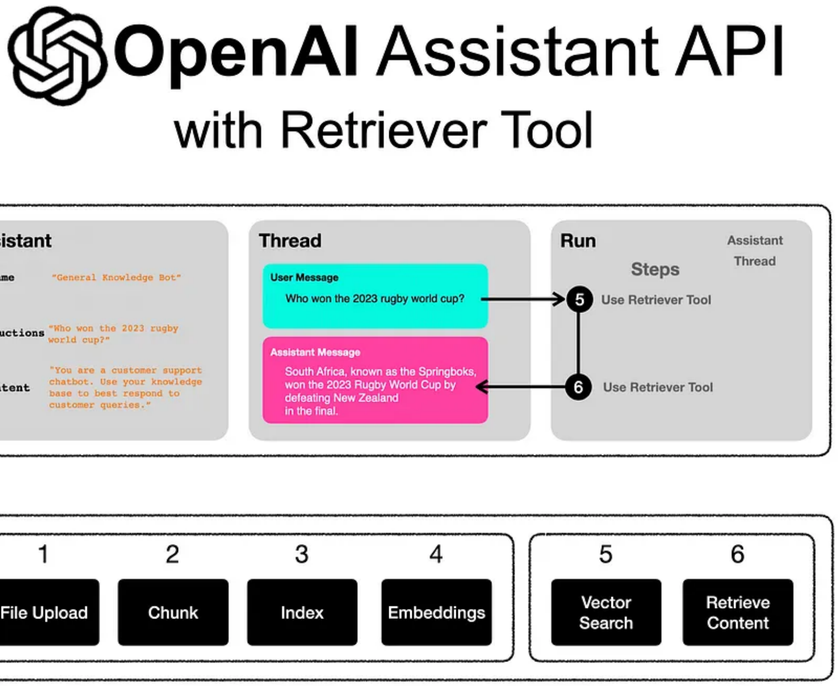
Chatbot using OpenAI's Assistant API

Introduction

This document provides an overview of a customer support chatbot designed to answer queries using a specific PDF file as a knowledge base. The chatbot is developed using OpenAI's GPT-3.5 model and leverages the latest advancements in natural language processing to provide accurate and contextually relevant responses.

Design



name "General Knowledge Bot"

instructions "Who won the 2023 rugby world cup?"

content "You are a customer support chatbot. Use your knowledge base to best respond to customer queries."

Who won the 2023 rugby world cup?

South Africa, known as the Springboks, won the 2023 Rugby World Cup by defeating New Zealand in the final.

The chatbot system is designed using Python and involves integration with OpenAI's API. The process is outlined below in a tabular format and compares it with chat completion API:

Comparison: Assistant API vs Chat Completion API

Step	Assistant API	Chat Completion API
1. Initial Setup	Obtain API keys, set up authorization.	Obtain API keys, set up authorization.
2. Document Upload	Uses Retrieval tool for accessing and processing information from documents.	Focused more on chat inputs than on document upload.
3. Define Prompt/Start Conversation	Create a text prompt or start a conversation in a new Thread.	Initiate with a conversational message.
4. Choose Model	Select an appropriate model (e.g., GPT-3, Codex).	Uses a model optimized for chat interactions.
5. API Request/Send Message	Send a request to Run the Assistant on a Thread with Messages.	Send a message maintaining conversation context.
6. Processing	Assistant uses configuration and Messages in Thread to perform tasks.	Generates a reply considering the ongoing conversation.
7. Receive Response	Assistant appends new Messages to the Thread as output.	Receive a conversational response relevant to the chat.
8. Post-Processing/Maintain Context	Interpret and integrate responses; review Run Steps for insights.	Use the response in the conversational interface, keeping context.

Implementation

To set up and run the chatbot, follow these steps:

1. Create a Python virtual environment and activate it:

```
python3 -m venv venv

. venv/bin/activate
```

2. Install necessary packages:

```
pip install openai python-dotenv
```

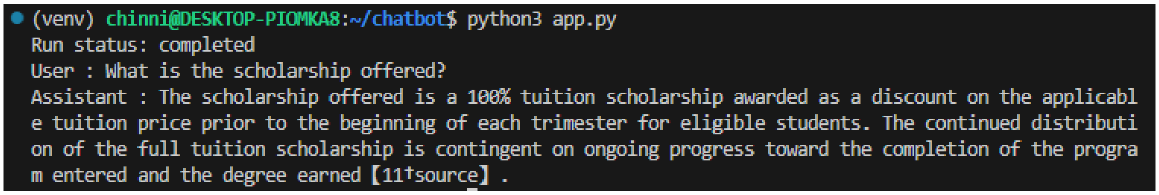
3. Run the application:

```
python3 app.py
```

The python code(app.py) can be found on my github.

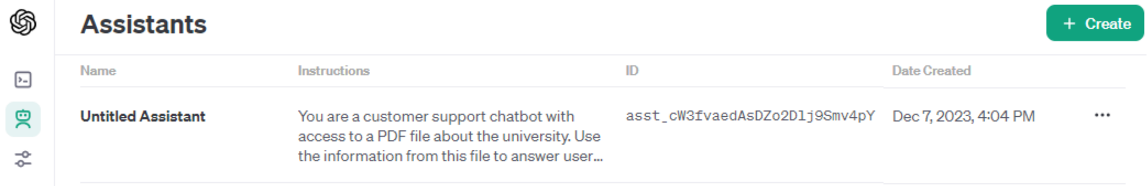
Testing

Here's the result:



```
● (venv) chinni@DESKTOP-PIOMKA8:~/chatbot$ python3 app.py
  Run status: completed
  User : What is the scholarship offered?
  Assistant : The scholarship offered is a 100% tuition scholarship awarded as a discount on the applicabl
  e tuition price prior to the beginning of each trimester for eligible students. The continued distributi
  on of the full tuition scholarship is contingent on ongoing progress toward the completion of the progra
  m entered and the degree earned 【11†source】 .
```

The assistant created here can also be tested on the OpenAI web interface.



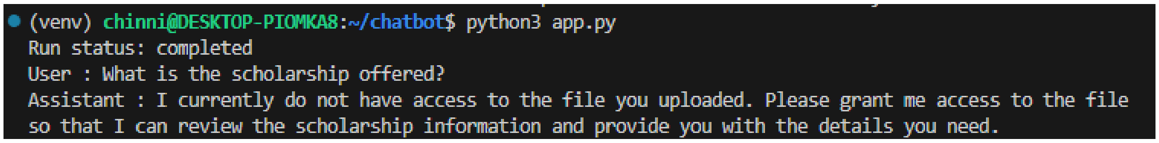
Name	Instructions	ID	Date Created
Untitled Assistant	You are a customer support chatbot with access to a PDF file about the university. Use the information from this file to answer user...	asst_cW3fvaedAsDZo2Dlj9Smv4pY	Dec 7, 2023, 4:04 PM

Note:

- We can also create and test the assistant using the OpenAI Playground web interface. Here's a tutorial pdf.

- An assistant created in the web interface can be used in our project with the assistant ID, eliminating the need to recreate the assistant and upload the file using Python.

- In the current implementation, it's important to note that the assistant API is in beta mode. As a result, there are occasions(as shown) when the assistant may respond indicating that it does not have access to the uploaded document for answering queries. This is a known issue and is expected to be resolved as the API matures and is further developed by OpenAI.



```
● (venv) chinni@DESKTOP-PIOMKA8:~/chatbot$ python3 app.py
  Run status: completed
  User : What is the scholarship offered?
  Assistant : I currently do not have access to the file you uploaded. Please grant me access to the file
  so that I can review the scholarship information and provide you with the details you need.
```

Enhancement Ideas

Future enhancements can include:
- Improved error handling and logging for more robust operation.
- Integration with a user interface for easier interaction.
- Expansion of the knowledge base to cover a broader range of topics.

References

For more information, visit:
- OpenAI Assistants Overview: [OpenAI Assistants Documentation](https://platform.openai.com/docs/assistants/overview)