Project 1

can code communities

Take Home Exercise Review

Use the Assistants and Playground tabs on OpenAI to create a coding assistant that helps build a simple pong game.

Review Questions

1. How is the data formatted for fine-tuning?
2. What are some initial approaches before deciding to fine-tune a model?
3. What are three helpful features of assistants?

Project Overview:

This project aims to develop a chatbot using the OpenAI API implements function calling to connect the chatbot to external data sources. Students will be provided a list of libraries and APIs that can be easily attached to the chatbot to expand its ability.

Libraries and APIs

- Datetime
  - Python library that can return the current time and date.
- Psutil
  - Python library that can return the battery status of the device the script is running on.
- NewsAPI
  - News aggregation API that can return the current top headlines.
- OpenWeatherMap API
  - Weather API that returns current weather based on location.
- Wolfram Alpha Short Answers API
  - Answer engine that returns factual information.

Project Submission

Your project will be due on 7/12/2024

There will be a Teams assignment created that you will use to submit your project, you can use one of the following two options to submit your project:

1. You can upload the required files directly to Teams and then submit your project.
2. You can first upload the required files to a public GitHub repository, then link to your GitHub repository as your Teams assignment submission.

The required items can be seen below.

Project Objectives:

1. Create a conversational chatbot using the OpenAI Chat Completions API.
2. Create at least 3 functions that retrieve data from the list of libraries and APIs provided.
3. Create the tool list that describes these functions and the arguments necessary for the function in natural language.
4. Append the data returned from the called function to the message list with the role “tool” and make another API call for the final response.
5. Document the entire process for reproducibility and future reference.

Project Tasks:

1. Conversational Chatbot
    a. Make a loop where the prompt and response pairs are appended to the message list.
    b. Manage the message list based on the total tokens in the response object so that the chatbot doesn't hit the context limit.
2. Data Retrieval Functions
    a. Set up accounts and API keys for the chosen data source APIs.
    b. Use the python libraries and APIs within functions to return the desired data.

Project Tasks:

3. Function Description
    a. Detail the intended use of the functions in natural language so the model can understand when to use the functions properly.
    b. Determine what arguments (if any) are necessary for each function.
4. Function Calling Implementation
    a. Use the response object to call the function(s) chosen by the model.
    b. Append the function result to the message list for the model to generate the natural language response.

Project Tasks:

5. Documentation
    a. Create detailed documentation explaining the generation steps, assumptions, and decisions.
    b. Include comments in the code to make it more understandable and maintainable.
6. Testing and Validation
    a. Test the function calling with multiple prompts that invoke different (or multiple) functions to be called.
    b. Validate the results against the expected outcomes.

Items for Submission:

1. A documented generation process, including code comments and explanations.
2. A script for the conversational function calling chatbot.
3. Any additional documentation or instructions for using the chatbot.

Questions?

Project Work Time