From Hallucinations to Real Insights: Supercharging GPT 3.5 Turbo with External APIs

OpenAI’s ChatGPT has been progressing consistently and releasing new features such as voice, image, and plug-in capabilities. I’ve found these developments to be helpful, but unfortunately there are some limitations with these features and their accessibility. The plug-in capability is only available with a ChatGPT Plus or Enterprise membership, which restricts the use of such a powerful tool. Even though OpenAI is always announcing lower token rates, and giving users free options to use AI, their new features are usually only released for their paid memberships. ChatGPT Plus costs $20/month and their Enterprise tier is even more expensive. My goal in this project was to create a plug-in like feature for the GPT 3.5 Turbo model, which is practically free relative to the monthly subscriptions.

The release of the plug-ins was arguably the biggest feature OpenAI has released. Conceptually, ChatGPT is a chatbot with inconsistent quality responses. The term “hallucinating” in regards to AI models refers to when the AI starts to make up facts and statements that don’t match reality. This seems to be the effect of the word-by-word generation architecture of the model. There are also cutoff dates for training data that these models use. Even GPT4, which is the newest released model that is only available with a membership, has a cutoff date about 2 years ago (January 2022). This adds to the inconsistency of responses, especially relating to real-time data or even the near past data.

GPT’s Plug-in feature allows a way to eliminate these problems. With the use of APIs, users are able to retrieve real-time and verified data from other sources. Plug-ins use these APIs to get specific data determined by the AI based off the context of the prompt, and then incorporate the knowledge gained into the conversational element of GPT.

My current project implements the OpenWeatherMap and Wolfram Alpha APIs, which is just the start of the possibilities to round out the AI model’s capabilities. OpenWeatherMap retrieves real-time weather information and Wolfram Alpha gets the computational answers to complex or fact-based questions. These APIs are integrated into GPT using OpenAI’s function calling feature, which is readily accessible. Basically, these functions are described in formatted JSON data that gives context to the AI model for why and when these functions would be used. This data is given to the model when retrieving the response. Alongside the function description data, a function calling parameter is set to “auto”. This allows the model to decide whether or not to use the functions provided, and then which function to call. So GPT can generate responses to queries the model is comfortable with answering, but when it isn't, it will defer to calling the respective functions with the proper arguments.

Outside of APIs, which is web-based data retrieval method, there are even a couple of simple python library packages that can help supply helpful information to the model as well. I used the datetime library to get the current time and date, and the psutil library to get the status of the device's battery. These functions run computations on the local machine which is faster and more efficient than an API call to get the same information.

For this project, I succeeded in creating a chatbot that uses external APIs to expand and verify its knowledge base while still maintaining the conversational ability. This is the first step of development though. To improve this project, I'll be focusing on adding other APIs and python libraries that can perform tasks outside of the scope of the model. Even though the cost of using GPT 3.5 Turbo is tenths of a cent per a thousand tokens, adding more contextual functions for the model's use will raise the amount of tokens used with every prompt. So, minimizing the cost while maximizing the ability will be implemented within the upcoming versions of this project.