OpenAI Module

4 min

To harness the power of language models like GPT, we'll start by diving into the API's OpenAI Python module. The OpenAI module provides classes, tools, and resources that allow developers to interact with and integrate GPT models into their applications.

Within Python, the OpenAI module is called openai and allows us to initiate requests to the GPT models with ease.

OpenAI Class

Within the openai module exists a class called OpenAI. This class provides functionality designed to streamline interactions with the OpenAI GPT models. We can control the generation of responses and tailor the behavior of the language model to suit the requirements of our applications.

To begin using the OpenAI class, we can import it from the openai library:

from openai import OpenAI

Once we have imported the OpenAI class, we can instantiate it and save it to a variable called client:

client = OpenAI()

Our client variable will be used later for creating prompts and engaging in conversations with the AI model.

Models

The main component of generative AI is the GPT model used to receive the prompt and create a response. The two key models we will focus on are:

- **GPT-3.5 Turbo**: A cost-effective option, GPT-3.5 Turbo delivers high-quality content generation and performance for most applications, with an input (user prompt) and output (model reply) token limit of 4096.
- **GPT-4 Turbo**: GPT-4, OpenAI's latest release as of this writing, boasts enhanced power, correctness, and precision over GPT-3.5 Turbo. With a maximum input and output token limit of 128,000, it retains historical context, enabling more accurate responses and content analysis. However, it does come with an increase in price and slower performance compared to GPT-3.5 Turbo.

So what are *tokens*. Tokens are individual characters or words that make up the prompts and generated replies. If the conversation prompt or response exceeds this token limit, it may be necessary to shorten or reword the prompt.

OpenAI API Authentication

The OpenAI API requires the use of API keys that are used for authentication. The exercises within this lesson provide a pre-supplied authentication key for seamless access to the API. If you intend to use the OpenAPI API outside of this course, we recommend consulting with your organization's IT and Security department to determine the best practices for securing and managing API keys that are in accordance with your organization's security protocols.

Instructions

1. Checkpoint 1 Passed

1.

Import the OpenAI class from the openai module.

⍰ Use the following syntax:

```
from module_name import ClassName
```

⍰ Checkpoint 2 Passed

2.

Instantiate the OpenAI class and store it within a variable called client.

Use the following syntax:

```
variable_name = ClassName()
```

script.py

```
from openai import OpenAI


client = OpenAI()
```