

# Purpose

In this notebook I will illustrate how you can use the ChatGPT API to build a question and answer system.

## Setup

This cell imports the necessary modules and libraries to use the OpenAI API. Specifically, it imports `os`, `openai`, and `load_dotenv` from the `dotenv` library.

`os` is a module that provides a way of interacting with the operating system. `openai` is the Python client for the OpenAI API, which allows you to interact with OpenAI's GPT models. `load_dotenv` is a function that loads environment variables from a `.env` file into `os.environ`, which allows you to access them in your code.

```
import os
import openai
from dotenv import load_dotenv
```

This code block imports the `load_dotenv()` function from the `dotenv` library, which loads environment variables from a `.env` file into `os.environ`. It then sets the `openai.organization` and `openai.api_key` variables to the values of the `OPENAI_ORGANISATION` and `OPENAI_API_KEY` environment variables, respectively, using the `os.getenv()` function. This allows the OpenAI API to authenticate the user and authorize access to the API.

```
# Make your authentication keys available to the library
load_dotenv()
openai.organization = os.getenv('OPENAI_ORGANISATION')
openai.api_key = os.getenv('OPENAI_API_KEY')
```

## Example 1: Simple Q+A function

This function `prompt_gpt` takes in a string `content` and an optional string `model` as input parameters. It then creates a list of messages to send to the OpenAI GPT model, with the first message being a system message and the second message being the user's message. The function then creates a completion using the OpenAI API, passing in the `model`, `temperature`, `max_tokens`, and `messages` parameters. Finally, the function appends the response from the model to the messages and returns the completion object.

```
def prompt_gpt(content: str, model: str = "gpt-3.5-turbo") -> openai.ChatCompletion:
    # create messages to send to the model
    messages = [
        {"role": "system", "content": "You are a teaching assistant helping students w"},
        {"role": "user", "content": content},
    ]
    # create a completion
    completion = openai.ChatCompletion.create(
        model=model,
        temperature=0,
        max_tokens=1000,
        messages=messages
    )
    # add the response from the model to the messages
    messages.append(completion['choices'][0]['message'])
    return completion
```

Here we test the function by passing in a string `content`.

```
completion = prompt_gpt("Hello Sir, can you help me with the concept of data architecture?")
print(completion['choices'][0]['message']['content'])
```

Of course! I'd be happy to help you understand the concept of data architecture.

Data architecture refers to the design and organization of data within an organization. It is

There are several key components of data architecture:

1. Data sources: These are the various systems, databases, and applications that generate or
2. Data storage: This refers to the physical or virtual storage of data. It can include data

3. Data integration: Data integration involves combining data from different sources and formats.
4. Data modeling: Data modeling is the process of creating a conceptual representation of data.
5. Data governance: Data governance refers to the policies, processes, and standards that ensure data is managed properly.
6. Data security: Data security involves protecting data from unauthorized access, use, or disclosure.

By establishing a well-designed data architecture, organizations can ensure that data is accessible, secure, and of high quality.

I hope this explanation helps clarify the concept of data architecture for you. Let me know if you have any further questions!

## Example 2: A simple Q+A chatbot

This code creates a chatbot that uses the OpenAI API to generate responses to user input.

The code starts by creating a list of messages with a system message. It then enters a loop that continues until the user types 'exit'. Within the loop, the code prompts the user to enter a message and adds the user's message to the list of messages. It then creates a response using the OpenAI API, passing in the list of messages as input. Finally, the code prints the response from the assistant and adds the assistant's message to the list of messages.

The chatbot uses the OpenAI GPT-3.5-turbo model to generate responses. The `temperature` parameter controls the randomness of the responses, with a value of 0 indicating that the model should always choose the most likely next word. The `max_tokens` parameter controls the length of the response, with a value of 1000 indicating that the response should be no longer than 1000 tokens.

```
# create a list of messages with a system message
messages = [
    {"role": "system", "content": "You are a teaching assistant helping students with their homework."}
]

# loop until the user types 'exit'
while True:

    # get user input
    prompt = input("Please enter your message. Type 'exit' to quit.")
    print(f" --- \n You: {prompt} \n")

    # if the user types 'exit', break out of the loop
    if prompt == "exit":
```

```

        break

# add the user's message to the list of messages
messages.append({"role": "user", "content": prompt})

# create a response using the OpenAI API
response = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    temperature=0,
    max_tokens=1000,
    messages=messages
)

# print the response from the assistant
print(f" --- \n Assistant: {response['choices'][0]['message']['content']} \n")

# add the $assistant's message to the list of messages
messages.append(response['choices'][0]['message'])

---
You: Hi, can I ask a question?

---
Assistant: Of course! I'm here to help. What's your question?

---
You: I want to know more about a topic that was discussed during class.

---
Assistant: Sure, I'd be happy to help you with that. Could you please let me know which topi

---
You: It was about designing a good research question.

---
Assistant: Designing a good research question is an important step in any research project.

1. Be specific: Your research question should be clear and specific. Avoid broad or vague qu
2. Be focused: Narrow down your research question to a specific aspect or problem within your
3. Be relevant: Ensure that your research question is relevant to your field of study and al

```

4. Be feasible: Consider the resources, time, and access to data or participants that you have.
5. Be answerable: Your research question should be answerable through research methods and data.
6. Be interesting: Choose a research question that you are genuinely interested in. This will help you stay motivated.

Remember, your research question may evolve as you conduct your literature review and gain a better understanding of the topic.

I hope these tips help you in designing a good research question. If you have any specific questions, feel free to ask!

---

You: exit

Can we make the student assistant a bit more sophisticated?

```
# create a list of messages with a system message
messages = [
    {"role": "system", "content": "You are a teaching assistant helping students with their research."}
]

# loop until the user types 'exit'
while True:

    # get user input
    prompt = input("Please enter your message. Type 'exit' to quit.")
    print(f" --- \n You: {prompt} \n")

    # if the user types 'exit', break out of the loop
    if prompt == "exit":
        break

    # add the user's message to the list of messages
    messages.append({"role": "user", "content": prompt})

    # create a response using the OpenAI API
    response = openai.ChatCompletion.create(
        model="gpt-4",
        temperature=1,
        max_tokens=1000,
        messages=messages
    )
```

```
# print the response from the assistant
print(f" --- \n Assistant: {response['choices'][0]['message']['content']} \n")

# add the $assistant's message to the list of messages
messages.append(response['choices'][0]['message'])
```

---

You: Hello, can I ask a question?

---

Assistant: Of course! What question do you have on your mind?

---

You: It's about a topic from the previous class.

---

Assistant: Absolutely. Could you share more details about the specific topic or problem you

---

You: So, the instructor pointed out that we should critically assess the resources we are us

---

Assistant: That's an important point to ask about. Let's consider what factors might indicat

---

You: Well, maybe who the author is?

---

Assistant: Yes, that's a good start! Understanding the author's background and qualification

---

You: What the purpose of the resource is?

---

Assistant: Indeed, understanding the purpose or motive behind a resource can offer great ins

Now, imagine you also had to evaluate the quality of the information itself in the source. W

---

You: Whether it relates to my research

---

Assistant: Absolutely. Relevance to your research is key. But in addition to relevance, do y

---

You: Ah, for instance whether I can check the validity of what is mentioned in the informat

---

Assistant: Absolutely, that's a core facet. Being able to verify the information against oth

Now, having discussed the author, the purpose, the relevance, and the verify-ability, can yo

---

You: So, I should check the background of the author, the purpose of the published informat

---

Assistant: That's an excellent summary! You've covered all the key points in critically ass

---

You: exit