**Lab Exercise 10- Zero-Shot, Few-Shot, and Chain-of-Thought Prompting with Azure OpenAI**

**Prerequisites**

1. **Azure Subscription**: Ensure you have an Azure account with access to the OpenAI service.
2. **OpenAI API Key**: Get your API key and endpoint from the Azure OpenAI resource.
3. **Environment Setup**: Ensure you have the necessary libraries installed. In Google Colab, you can use the following command:

!pip install openai

1. **Enable the Azure OpenAI SDK**: In the Azure portal, make sure to enable the OpenAI resource.

**Step 1: Set Up Your Azure OpenAI API Key**

First, you will need to set up your API key and endpoint in the Google Colab environment.

import os

# Set up your API key and endpoint

os.environ["AZURE\_OPENAI\_API\_KEY"] = "your\_api\_key"

os.environ["AZURE\_OPENAI\_ENDPOINT"] = "your\_endpoint" # e.g., "https://your-resource-name.openai.azure.com/"

**Step 2: Import Required Libraries**

import openai

import json

**Step 3: Initialize the OpenAI Client**

Set up the OpenAI client with your Azure credentials.

# Initialize OpenAI client

openai.api\_type = "azure"

openai.api\_key = os.getenv("AZURE\_OPENAI\_API\_KEY")

openai.api\_base = os.getenv("AZURE\_OPENAI\_ENDPOINT")

openai.api\_version = "2023-05-15" # Use the appropriate version

**Step 4: Define a Function to Call the OpenAI API**

Create a function to send a request to the OpenAI API.

def query\_openai(prompt, model="gpt-35-turbo", temperature=0.7):

response = openai.ChatCompletion.create(

engine=model,

messages=[

{"role": "user", "content": prompt}

],

temperature=temperature,

max\_tokens=150 # Adjust as necessary

)

return response.choices[0].message["content"].strip()

**Step 5: Zero-Shot Prompting**

In Zero-Shot prompting, you provide a task without any examples.

# Zero-Shot Prompting

zero\_shot\_prompt = "What are the benefits of regular exercise?"

zero\_shot\_response = query\_openai(zero\_shot\_prompt)

print("Zero-Shot Response:")

print(zero\_shot\_response)

**Step 6: Few-Shot Prompting**

In Few-Shot prompting, you provide a few examples along with the task.

# Few-Shot Prompting

few\_shot\_prompt = (

"Translate the following English sentences to French:\n"

"1. Hello. -> Bonjour.\n"

"2. How are you? -> Comment ça va?\n"

"3. What is your name? ->"

)

few\_shot\_response = query\_openai(few\_shot\_prompt)

print("\nFew-Shot Response:")

print(few\_shot\_response)

**Step 7: Chain-of-Thought (CoT) Prompting**

In Chain-of-Thought prompting, you encourage the model to think step-by-step.

# Chain-of-Thought Prompting

cot\_prompt = (

"Calculate the following: If I have 3 apples and I buy 2 more, "

"how many apples do I have in total? Please show your work."

)

cot\_response = query\_openai(cot\_prompt)

print("\nChain-of-Thought Response:")

print(cot\_response)

**Step 8: Execute the Code**

Run all the cells in your Google Colab notebook to see the results of each prompting technique.

**Conclusion**

In this lab exercise, you learned how to implement Zero-Shot, Few-Shot, and Chain-of-Thought prompting techniques using Azure OpenAI in Google Colab. Each technique can be beneficial in different scenarios, allowing you to tailor your interactions with the language model effectively.