AI Exercises

Exercise 1: Basic Interaction with GPT Model

Objective:

Familiarize students with setting up the OpenAI Python client, connecting to a GPT model, and handling responses.

Tasks:

- Setup Environment: Install the OpenAI Python client and set up your API key.
- Connect to GPT-4: Write a Python script to connect to the GPT-4 model.
- Create a Prompt: Develop a prompt asking for a summary of the latest advancements in AI.
- Retrieve and Parse Response: Send the prompt to the model and parse the response to extract and print the summary.

Deliverables:

- Python script that performs the above tasks.
- A text file containing the retrieved summary.

AI Exercises

Exercise 2: Advanced Prompt Engineering

Objective:

Teach students how to craft complex prompts that generate structured responses.

Tasks:

- Prompt Design: Create a prompt that instructs the AI to generate a list of 5 innovative business ideas based on AI technologies.
- Interaction with GPT-4: Use the OpenAI Python client to send your prompt to the GPT-4 model.
- Response Handling: Parse the AI s response to separate each business idea into its own section.

Deliverables:

- Python script for sending prompts and handling responses.
- A report containing the structured list of business ideas.

AI Exercises

Exercise 3: Analyzing Sentiment with AI

Objective:

Enable students to use the AI model to perform sentiment analysis on user reviews.

Tasks:

- Data Collection: Collect or create a dataset of at least 10 short product reviews.
- Sentiment Analysis Prompt: Develop a prompt that asks the AI to analyze the sentiment of each review and classify it as positive, negative, or neutral.
- Processing Responses: Use the OpenAI Python client to send each review to the GPT model and process the sentiment classifications.

Deliverables:

- Python script for analyzing sentiments.
- A CSV file containing each review and its corresponding sentiment classification.

AI Exercises

Exercise 4: Summarizing Web Articles Using Langchain

Objective:

Teach students how to integrate the Langchain library with web scraping techniques to summarize online articles.

Tasks:

- Setup Environment: Install the necessary Python libraries including langchain, requests, and beautifulsoup4.
- Web Scraping: Write a function to fetch the content of the article from the provided URL. Use requests to get the web page and BeautifulSoup from beautifulsoup4 to parse the HTML and extract the article text.
- Langchain Integration: Utilize the Langchain library to connect to an AI model capable of summarizing text. Input the extracted article text to the model and get a summary.
- Output the Summary: Display the original article URL and its summary on the console.

Deliverables:

- Python script that takes a URL as input and outputs a summary of the article.
- A sample output showing the URL and its summary for at least three different web articles.

AI Exercises

Exercise 5: Story Creation with AI Model Chaining using Langchain

Objective:

Teach students how to use sequential AI processing with the Langchain library to create children's stories.

Tasks:

- Setup Environment: Ensure the langchain library is installed along with other dependencies.
- Story Idea Generation: Input a topic for a children s story. Use the first AI model to generate multiple story ideas or plots based on this topic.
- Story Development: Select one of the generated story ideas. Use the second AI model to elaborate this idea into a full children's story.
- Display and Save Results: Print and save the original topic, chosen story idea, and the complete story.

Deliverables:

- Python script that implements the AI model chaining to create a children's story.
- A text file containing the topic, chosen story idea, and the full story.

AI Exercises

Exercise 6: Monitoring AI Story Creation Chain in Langsmith

Objective:

Teach students how to monitor and visualize AI model interactions using Langsmith, an environment that supports the Langchain library.

Tasks:

- Setup and Configuration: Sign up for a free Langsmith account. Install the langchain library and ensure Langsmith is set up according to the provided documentation. Configure Langsmith settings to track the Langchain operations.
- Implement the Story Chain: Implement the story generation chain (topic to ideas to full story) as developed in the previous assignment. Integrate Langchain operations with Langsmith monitoring to track each API call and its response.
- Monitor and Analyze: Use Langsmith to monitor the chain s execution, focusing on how each input (topic and selected idea) is processed by the AI models. Analyze the decision-making process visible in Langsmith, noting any interesting patterns or unexpected behaviors.
- Report Findings: Write a brief report detailing the process, including screenshots from Langsmith that show the chain of model interactions. Discuss any insights gained from monitoring the AI s decision-making process.

Deliverables:

- Python script configured to work with Langsmith for tracking.
- A report including setup steps, screenshots from Langsmith, and analysis of the AI interactions.

AI Exercises

Exercise 7: Creating Streamlit Apps

Objective:

Create Streamlit applications for previously developed exercises.

Tasks:

- Create Streamlit app for Exercise 4.
- Create Streamlit app for Exercise 5.
- Create Streamlit app for Exercise 6.

Deliverables:

- Python script configured to work with Streamlit apps.