XevenSkills

Assignment #03

Hope to Skills

Free Artificial Intelligence Advance Course

Instructor: Irfan Malik, Dr. Sheraz

Submission:

- Make a Google Collab notebook to implement this assignment.
- In case you face difficulty in creating the Google Collab Notebook Follow these **Steps**
- Submit a **.ipynb** file detailing all the information. No other format will be accepted
- Submission file should be named as **Assignment_03_StudentName.ipynb**
- Deadline for this Assignment is **Friday 01-03-2024**.
- Strictly follow the submission deadline.
- Make Submission in the **Assignment-03** Google Form and press the submit button.
- Click **here** to submit the Assignment

What you will learn

- How to create Google Colab Notebook from scratch.
- Basic understanding and API usage of OpenAI.
- Basic understanding and hands on experience of HuggingFace models.

Solve the Following Tasks

Question 1: Explain the followings:

- Define the context window in the context of language models.
- Explain why the model's context window is typically kept small.
- Identify and elaborate on which OpenAI language model boasts the largest context window and provide a detailed explanation.

Question 2: Please discuss your understanding on the following topics:

- Main challenges and limitations associated with training and deploying and how do researchers and developers address these challenges to ensure model reliability and performance?
- Explain your understanding on Embeddings and Model Fine Tuning

Question 3: Design a sentiment analysis application using OpenAI's models to determine the sentiment of a given text (positive, negative, or neutral)?

Question 4: Design a Python program that compares two text inputs and determines their similarity using the OpenAI API.

Instructions:

- Prompt the user to input two texts for comparison.
- Utilize the OpenAI GPT-3.5-turbo language model to analyze the similarity between the provided texts.
- Display the similarity score to the user, indicating how similar the two texts are.
- Handle errors gracefully, such as empty inputs or API request failures.
- Ensure the program is properly documented and easy to understand.
- Optionally, provide additional information to the user about the comparison process or
- the meaning of the similarity score.

Question 5: Develop a Python program that generates a blog post based on a user-provided topic. The program should use OpenAI's API to create a complete blog post including a title, a description (approximately 300 words), related keywords, SEO meta title, SEO meta description, and a corresponding image. The program should interact with the user through the input() function for topic input and display the generated content within the Colab notebook.

Hints:

- *Review OpenAI's API documentation, and use the appropriate models for the task.*
- *Practice prompt engineering to guide the AI for each specific content piece (title, description, keywords, meta title, and description).*

- *For image generation, describe the kind of image that would be suitable for the blog post and visualize it accordingly.*
- *Implement robust error handling for API interactions.*

Question 6: What is Hugging Face and why do we briefly explore it?

- Explore Transformers
- Explore pipeline
- Explore Hugging face API

Question 7: Build a simple text sentiment analyzer that categorizes sentences as positive,negative, or neutral.
Hint: *Pick any pre-trained HuggingFace sentiment analysis model from the Hugging Face.*

Question 8: You have a corpus of news articles containing information about different companies and their locations. Use HuggingFace model, develop a pipeline that extracts the names and locations of these companies from the text.

Question 9: Create a text generator that produces creative text continuations based on your input prompts.
Hint: *Select a pre-trained model suited for text generation from the Hugging Face*

Question 10: Use any one hugging face model in your project. - Write a python code that uses HuggingFace tokenizers library to tokenize a given sentence using the BERT tokenizers.

Notes for Students:

- Each question is of **10 marks**.
- Please use proper comments and formatting in your code. Bad formatting or no comments will result in marks deduction.
- Any other file format except ".ipynb" will not be accepted and will result in 0 marks.