Assignment

AI-Powered Text Completion Project Report

Introduction

The objective of this project was to build and experiment with a simple AI-powered text completion application using Generative AI. The aim was to understand how these models generate text, explore prompt design, adjust parameters, and evaluate the model's capabilities and limitations. This project provided hands-on experience interacting with a pre-trained AI model through an API, allowing us to see how advanced language models can process and respond to human-like prompts.

Setup and Approach

For this project, I chose **OpenAI's GPT-3.5-turbo** model as my generative AI provider. I obtained an API key from the OpenAI platform and securely integrated it into my local Python script.

The environment was set up using Python, and the openai library was installed using pip. The final application, named text_completion_app.py, was designed to accept user input from the terminal, send it to the model, and display the generated response in a readable format. The app also included error handling for invalid API keys and basic input validation to ensure smooth execution.

Although the code allowed adjusting parameters like temperature and max tokens dynamically, in my final testing I used default settings for simplicity and consistency across outputs.

Experimentation and Prompts

To test the application, I used five diverse prompts covering creative, explanatory, poetic, and conceptual scenarios. Below are the prompts and summaries of their outputs:

Prompt 1: Once upon a time, there was a robot who loved music.

The model generated a creative short story about a robot discovering music and sharing joy with others. The output was coherent and imaginative, showcasing the model's strength in narrative generation.

Prompt 2: Explain photosynthesis to a 10-year-old.

The model provided a clear, child-friendly explanation of photosynthesis, comparing it to plants "eating sunlight" to make their own food. The language was simple and accessible, which was perfect for a young audience.

Prompt 3: Write a haiku about the ocean.

The output was a concise and poetic haiku capturing the essence of waves and the ocean breeze. Even though I did not change the temperature and token settings as initially planned, the response was creative and aligned well with the form of a haiku.

Prompt 4: Describe what the future of space travel might look like.

The model offered an optimistic and visionary description of future space exploration, including possibilities like colonizing Mars and building advanced space habitats. The response was imaginative and inspiring.

Prompt 5: Explain recursion to me like I'm five.

The model gave a simple analogy, comparing recursion to a story inside a story or looking at a mirror reflecting another mirror. It effectively simplified a complex concept into terms understandable for a young child.

Evaluation and Analysis

The outputs across all prompts were relevant and highly coherent. The model performed exceptionally well in narrative and creative tasks (e.g., storytelling and poetry), as well as in simplifying technical topics for non-technical audiences.

In this experiment, I used default settings (temperature 0.7 and max tokens 150) for most prompts. Even without explicit parameter adjustments, the model displayed a good balance of creativity and focus.

Strengths

- Excellent at generating creative stories and poetry.
- Strong ability to explain complex concepts simply.

- Coherent, contextually appropriate, and grammatically correct outputs.

Limitations

- Sometimes responses can be verbose or slightly repetitive.
- Without manual adjustments to temperature or token limits, some outputs may be longer than needed or less focused.

Reflection

This project highlighted the impressive capabilities of generative AI for creative and explanatory tasks. While the model excels in producing coherent and engaging text, careful prompt design and parameter tuning are essential to control tone, style, and length.

Moving forward, possible improvements could include implementing stricter content filtering, automatic summarization for long outputs, or allowing real-time parameter tweaking through the app interface.

Overall, this project deepened my understanding of generative AI models and strengthened my skills in integrating APIs into functional Python applications.