Gift Assistant

Team 3, Nov 15, 2023

Introduction

Introducing Gift Assistant, a festive solution powered by the llama-2-70b-chat. Our objective is to simplify Christmas gift selection. Users input recipient's details, and the AI crafts personalized recommendations. With the festive season just around the corner, we have chosen this project to assist in the process of selecting and giving gifts.

Background

This part should briefly cover the theoretical foundation of generative AI models, discuss relevant existing applications, and highlight how the project aligns with or diverges from established practices.

Methodology

The LLM used for this project is llama-2-70b-chat, a 70 billion parameter language model from Meta. The reason behind choosing this model is that we noticed it was the most efficient. According to the model webpage, it is designed to be less resource-intensive than other models, thus making it more available.

Application Design and Development

We built an interactive demo using the gradio python library. We implemented a simple form to gather inputs from the user and we passed them to the model API. For this purpose, we used the replicate python library to interact with the llama-2-70b-chat model hosted on the replicate website.

Experiments and Results

Having selected llama-2-70b-chat, which is a collection of pretrained and fine-tuned generative text models, we only needed to collect inputs from the user and use them to generate the proper prompt.
Since the model is hosted on replicate, we had the chance to interact with it directly on the website. That gave us the chance to run prompts, check the results and adjust the prompts accordingly.

Challenges and Problem-Solving

The primary challenge we faced was the limitation imposed on the free tier for all the cloud hosted LLMs we've explored, which limited our project to involve only one LLM.

Alternatively we could have utilized a local LLM for ourproject, but that was not feasible due to the fact that the computational requirements for large-scale models, such as the one we employed (llama-2-70b-chat), surpass the capabilities of our local systems, necessitating cloud-based solutions for efficient processing. Additionally, the extensive pre-training and fine-tuning requirements of these models often involve substantial computational resources that are more readily available in cloud environments.

Discussion

The application accomplishes its goal with good results. The response is well written and contains good suggestions for the user. With the design of the interface it is easy for us to control the user's input and make sure the model is receiving a well designed call.

The largest issue with the use of the application is the long time for response and the limits of using API. A free account using such API has a limited rate and resources so the time to receive a response is often long.

Reproducibility

We've included a README.md file on the github repository which includes the steps required to run the app locally. All the libraries and dependencies are listed in the requirements.txt file.

Conclusion

During the process of building this app, our group has overcome challenges mainly with model selection and API integration. The experience has given us a deeper understanding of the complexities and possibilities of using generative AI in real-world applications.

Future Work

For this project, we believe it would be beneficial to incorporate a visual representation of the gifts generated by the LLM. This could be accomplished by utilizing the API results to generate images through a text-to-image generative AI model.

References

Assistance in language refinement was provided by ChatGPT during the development of this report (OpenAI, personal communication, 2023, November 15).