Assignment # 4 Bonus

Generative AI Deadline: May 12, 2024

Instructions

Note: This is a bonus assignment. Those students can submit it who have already completed their three assignments. Its marks will be added/replaced in any of the three quizzes. There is a grace time of **2 hours after the submission deadline expires**. You must verify that your submissions are correct. Any submission received after this slack time will be considered late and NO marks will be awarded. There should be no comments provided on each line of code. You must provide detailed comments about functions that you create; that is enough to understand the function.

In case you are also doing a project related to RAG, you have to show a significant difference between this work and yours. For example, you can apply a different technology stack.

Deliverables:

1. Original Source file, including all code. (50% Marks)

2. **Technical Report.** The technical report must describe each experiment you did, and show the results in tables, visualizations, etc. All tables and figures must be discussed in detail. Explain each result, like why one model produced less accuracy, why training error was low, and validation was high, what caused this condition, etc. (50%)

ZIP File Submission

Put all your files inside one folder and name it according to the following convention:

RollNo_Name_Ass1.ZIP

You must upload this ZIP file

Objectives

The assignment requires the design and implementation of a web-based chat application using Retrieval Augmented Generation (RAG) techniques. This application will use the combination of an open source large language models (LLM) and structured data from data related to MS Thesis titles and abstracts provided in an excel sheet. The chat application should be able to generate insightful, contextually relevant information using both given data and LLMs.

This assignment provides an opportunity to explore cutting-edge AI technologies and apply them in a novel, practical context. By completing this task, you will gain hands-on experience in the burgeoning field of generative AI and retrieval-augmented systems. You are also free to use any technology stack, however, the recommended tools should be used.

Requirements:

1. Data Integration: Utilize the dataset provided, containing details of MS Theses completed at FAST. You will extract and use only the Thesis titles and abstracts and put this information in an open source vector database such as Qdrant or Faiss.

2. Technology Stack: For Retrieval Augmented Generation I am expecting you to employ libraries such as LangChain, LangSmith, or other relevant libraries to integrate the capabilities of an open-source LLM with the thesis data. I suggest using any available open-source LLM to power the generative aspect of the application.

3. GUI Development:

- Develop a web-based chat interface where users can interact in with the LLM. The chat application should be able to retrieve information from the vector database to provide contextually enriched responses based on the thesis data. Ensure the application can be deployed locally for demonstration purposes.

4. Evaluation

- Implement and document proper evaluation techniques to measure the performance of your application. Consider metrics such as response relevance, accuracy of information retrieval, user satisfaction, and system efficiency.

5. Documentation and Submission

- Provide a comprehensive report detailing the design, implementation, and evaluation of your application. Include code snippets, screenshots, and a step-by-step guide on setting up and testing the application locally.