Steps AI take home assessment

Objective:

Develop a question-answering system using RAG and HYDE techniques based on the Lex Fridman Podcast Transcript dataset.

Dataset:

We will use the "Lex Fridman Podcast Transcript" dataset, which contains comprehensive transcripts of conversations between Lex Fridman and his esteemed guests.

Source: Lex Fridman Podcast Transcript Dataset

Access: Publicly available on Kaggle.

Task Description:

The goal is to create an AI-based question-answering system that can provide accurate and contextually relevant answers based on the content of the Lex Fridman podcast transcripts. The system should:

- Retrieve relevant passages from the podcast transcripts based on the user's questions.
- Generate precise and contextually appropriate answers using the retrieved information.
- Enhance the quality of retrieved passages and generated answers using HYDE to improve the embedding representations.

Requirements:

LLM Models:

- If you have an API key, you can utilize the Open-AI GPT model.
- Alternatively, you are welcome to employ any other open-source Language Model (LLM), such as LLaMA-7B, Platypus, or Mistral.
- Additionally, Google offers free access to its Gemini model, albeit with certain limitations; please consult their documentation for further details.
- You have the freedom to choose any LLM that suits your needs.
- Your objective is to develop an HyDE-enabled Retrieval-Augmented Generation (RAG) based chatbot.

Data Preparation:

- Use the Lex Fridman Podcast Transcript dataset, which includes transcripts of various podcast episodes.
- Preprocess the data to ensure it is suitable for use in your models.

System Design:

- Implement RAG to retrieve relevant passages from the podcast transcripts based on user questions.
- Apply HYDE to enhance the embedding representations of the transcripts, ensuring high relevance and quality of retrieved content.
- Generate answers to user questions using the retrieved passages.

Implementation:

- Develop a Python-based application for the question-answering system.
- Use appropriate machine learning libraries and frameworks (e.g., PyTorch, TensorFlow, Hugging Face's Transformers, AWS, Azure, etc.).
- For GenAI models, you can use OpenAI, Google's Gemini, Anthropic, LLAMA, or any other GenAI model as deemed fit.
- Ensure the code is well-documented and modular.

Streamlit or Gradio Application:

Create a Streamlit or Gradio application that provides an interactive user interface for the question-answering system (optional). The application should allow users to:

- Input their questions.
- View the generated answers in an easy-to-read format.
- Explore the relevant passages from which the answers were derived.

Sample Streamlit or Gradio Application Workflow:

1. User Input: Allow users to enter their questions through a form.
2. Answer Generation: Trigger the RAG and HYDE-based model to generate answers based on the input questions.
3. Display Results: Present the answers on the screen, along with the relevant passages from the podcast transcripts.
4. Explore Passages: Provide an option to explore the retrieved passages that were used to generate the answers.

Deliverables:

Codebase:

- A well-documented code repository (GitHub or similar platform).
- Include clear instructions on how to set up and run the project.

Documentation:

- Detailed project report covering:
    - Problem statement and chosen solution.
    - Technical approach and implementation details.
    - Data sources and preprocessing steps.
    - Model training and evaluation methods.
    - Results and analysis.
    - Challenges faced and how they were addressed.
    - User manual for running the solution and interpreting the results.

Streamlit or Gradio based Application:

- A working Streamlit or gradio application for the question-answering system.
- Source code for the Streamlit/gradio app included in the code repository.
- Instructions on how to set up and run the Streamlit/gradio app.

Evaluation Criteria:

- Technical Proficiency: Accuracy and efficiency of the implemented solution.
- Innovation: Creativity and originality in applying RAG and HYDE techniques.
- Documentation Quality: Clarity, completeness, and professionalism of the documentation and report.
- Usability: User-friendliness and practicality of the final solution.
- Bonus Task: Quality and functionality of the Streamlit or Gradio application.

Submission:

- Submit your completed assignment via email to tech@stepsai.co within one week.
- Ensure that all deliverables are included and accessible.

The top 2 submissions will be awarded a reward of 5,000 INR each. We look forward to reviewing your submissions and exploring your innovative approaches to this AI challenge. Good luck!

Sample Questions for Testing the System
To test the question-answering system, consider using the following sample questions based on the Lex Fridman podcast transcripts:

1. AI and Ethics:
    - Question: "What does Lex Fridman discuss about the ethical implications of AI?"
2. Human-Robot Interaction:
    - Question: "What insights are shared on human-robot interaction?"
3. Technological Innovations:
    - Question: "What are some technological innovations mentioned in the podcast?"
4. Philosophical Perspectives:
    - Question: "How does Lex Fridman approach philosophical questions in his discussions?"
5. Guest Expertise:
    - Question: "What expertise do guests like Elon Musk bring to the podcast?"

Best,
Steps AI team.