Tarea de LLM

Lab Statement: Introduction to Creating RAGs (Retrieval-Augmented Generators) with OpenAI

Objective:
This lab is designed to introduce students to the fundamental concepts and practical implementation of Retrieval-Augmented Generators (RAGs) using OpenAI's tools and LangChain framework. By the end of the lab, students will have gained hands-on experience building and understanding RAGs, culminating in the delivery of two GitHub repositories showcasing their work.

Note 1: Use Pinecone as vector database. Use OpenAI for embeddings and LLM.
Note 2: About Pinecone:
https://python.langchain.com/docs/integrations/vectorstores/pinecone

---

1. Pre-Lab Preparation

- Familiarize yourself with the basic concepts of LangChain by following the tutorial available at **LangChain LLM Chain Tutorial**:
  https://python.langchain.com/docs/tutorials/llm_chain/
  This will provide the foundational knowledge needed for building advanced applications using LangChain.

---

2. Step-by-Step Guide to Build a RAG

- Follow the detailed tutorial at **LangChain RAG Tutorial**:
  https://python.langchain.com/docs/tutorials/rag/
  to construct a Retrieval-Augmented Generator.
  Ensure you understand how the retrieval mechanism integrates with the generative model to enhance responses with contextual information.

---

3. Implementation and Deliverables

- Implement the learned concepts from both tutorials to create your RAG project.
- Develop the code, ensuring it is well-documented and follows best practices.

---

4. GitHub Repositories

You must create **two** GitHub repositories:

Repository 1

- Contains the code and documentation for the basic LangChain LLM Chain tutorial.
- **Due date:** day of the workshop class.

Repository 2

- Contains the code and documentation for the RAG project.

---

5. README Requirements

Each repository must include a comprehensive README file that describes:

- The project’s architecture and components.
- Step-by-step instructions on how to install the dependencies and run the code.
- Screenshots or examples of the code in action, if applicable.

Assessment Criteria

- Completeness of code and adherence to the tutorials.
- Clarity and detail of the README documentation.
- Proper GitHub repository organization and presentation.

Submission

- Submit the links to your two GitHub repositories by the due date.