Alemeno

Assignment for Internship – AI/ML

Content Engine

In this assignment, you will be working on creating a Content Engine. This system analyzes and compares multiple PDF documents, specifically identifying and highlighting their differences. The system will utilize Retrieval Augmented Generation (RAG) techniques to effectively retrieve, assess, and generate insights from the documents

1. Setup

a) Backend Framework: Choose LlamaIndex or LangChain based on your comfort level and perception of which performs better for this use case.

- LlamaIndex: A flexible framework for creating custom retrieval systems.
- LangChain: A powerful toolkit for building LLM applications with a strong focus on retrieval-augmented generation.

b) Frontend Framework: Utilize Streamlit for building the user interface.

- Streamlit: An open-source app framework for Machine Learning and Data Science projects, allowing you to create interactive web applications easily.

c) Vector Store: Choose a vector store of your choice to manage and query the embeddings.

- Options include but are not limited to ChromaDB, Pinecone, Faiss, Milvus, Weaviate, etc.

d) Embedding Model: Select an embedding model for generating vectors from the PDF file content. Ensure the embedding model runs locally and is not exposed to any external services or APIs.

e) Local Language Model (LLM): Integrate a local instance of a Large Language Model for processing and generating insights. Ensure the LLM runs locally and is not exposed to any external APIs.

2. Initialization

You are provided three PDF documents containing the Form 10-K filings of multinational companies. These documents will serve as the basis for your comparison analysis. The documents are as follows:

1. Alphabet Inc. Form 10-K
2. Tesla, Inc. Form 10-K
3. Uber Technologies, Inc. Form 10-K

You will use these documents to implement and test your Content Engine. Your task is to retrieve the content from these PDFs, compare them, and answer queries highlighting the information across all documents.

Additionally, the end system should feature a chatbot interface where users can interact and obtain insights about information from the documents, compare numbers within these three documents, and more.

Sample Questions -

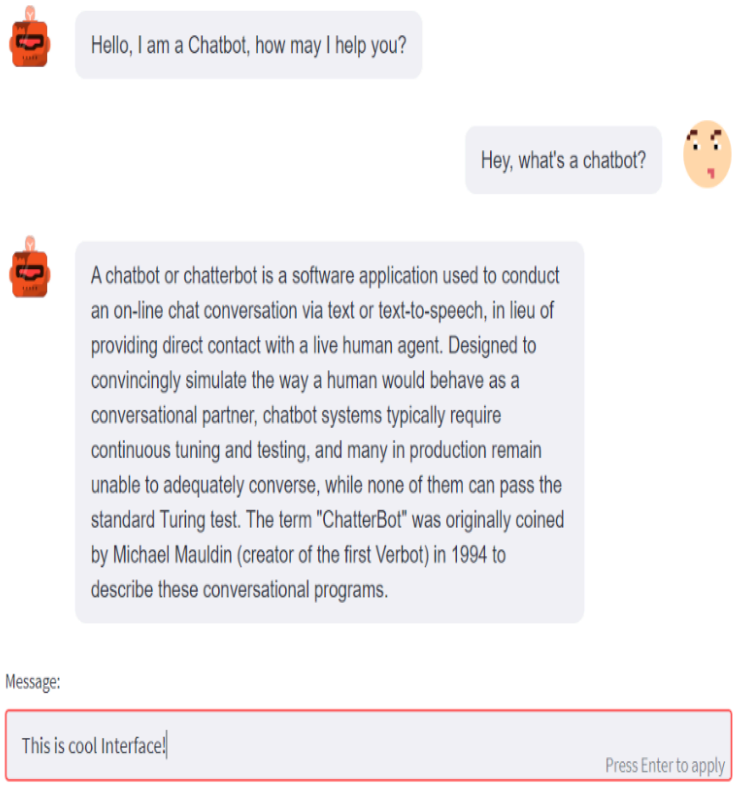
1) What are the risk factors associated with Google and Tesla?
2) What is the total revenue for Google Search?
3) What are the differences in the business of Tesla and Uber?

3. Development

- Parse Documents: Extract text and structure from PDFs.
- Generate Vectors: Use a local embedding model to create embeddings for document content.

- Store in Vector Store: Utilize local persisting methods in the chosen vector store.
- Configure Query Engine: Set up retrieval tasks based on document embeddings.
- Integrate LLM: Run a local instance of a Large Language Model for contextual insights.
- Develop Chatbot Interface: Use Streamlit to facilitate user interaction and display comparative insights.

4. Interface Sample



*UI is for reference only , feel free to build a similar UI using any tech stack of your choice

5. Expected Outcome/Guidelines

i. Design the overall architecture of the Content Engine, ensuring it is scalable and modular.
ii. A locally running LLM ensures data privacy and reduces dependency on external APIs.
iii. A chatbot interface for interacting with the system, allowing users to obtain insights and compare information across the documents.
iv. The assignment should be submitted within 4 days.
v. Please submit the GitHub link of the repository which should include the notebook used operations such as document processing, vector store ingestion, query engine development, and streamlit code.
vi. Ensure proper documentation and user guides are available.