Mr. Help Mate AI

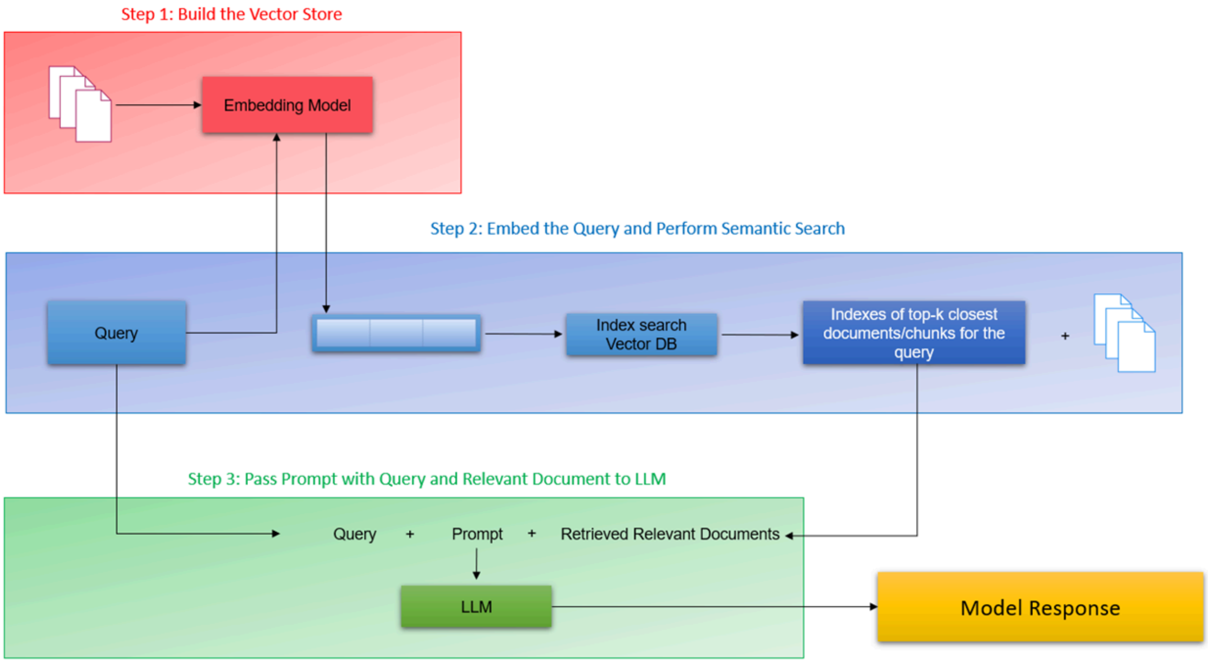
(A RAG Implementation)

Project Objective:

Build a project in the insurance domain, similar the project you saw in the "retrieval augmented generation" session. The goal of the project will be to build a robust generative search system capable of effectively and accurately answering questions from a policy document.

You will be using a single long life insurance policy document for this project.
The Goal of this project to implement a RAG project for insurance corpus.

HLD:


Step 1: Build the Vector Store
Embedding Model
Step 2: Embed the Query and Perform Semantic Search
Query
Index search Vector DB
Indexes of top-k closest documents/chunks for the query
+
Step 3: Pass Prompt with Query and Relevant Document to LLM
Query + Prompt + Retrieved Relevant Documents
LLM
Model Response

Project Descriptions:

Embedding Model:

We are embeding the pages in the dataframe through OpenAI's text-embedding-ada-002 model, and store them in a ChromaDB collection.That will be used for semantic search.

Caching :

We are also using in Cahcing to store the Retroved documents form Chrom DB.

Query :

User will send the the RAG System.

Top K- documents:

Frist user query is send to Cache if the results in found in Cache it will returns the results if not it will go to embedding vector and find the semtically relevant top k documents.

LLM:

LLM module is nothing but Chart Gpt. Once we received the top k document after semantic search we pass the top k documents as dataframe and query to ChatGPT to answe the correct results for for user query.

Cross-encoder:

We have used-cross-encoder to re-rank the semantic search results.

Attachments:

Semantic Search :

We have attached three screenshots for the top 3 results for semantic search for three custom queries.

Generative Search :

We have attached three screenshots for three custom query generative search results.

Code:

Jupyter Notebook developed for Mr Help Mate AI