Semantic Spotter Project Submission

1. Background

This project demonstrates "Build a RAG System" in the insurance domain using LangChain.

2. Problem Statement

The goal of the project is to build a robust generative search system capable of effectively and accurately answering questions from a collection of policy documents.

3. Document

The policy documents can be found in the specified directory.

4. Approach

LangChain is a framework that simplifies the development of LLM applications. It provides tools, components, and interfaces to construct LLM-centric applications. LangChain enables developers to build creative and contextually relevant content applications, supporting various providers such as OpenAI, Cohere, and Hugging Face.
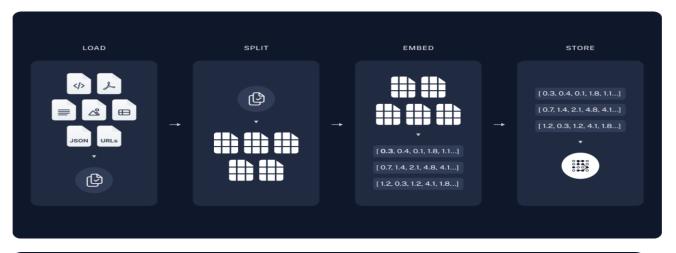
LangChain's design emphasizes composition and modularity. It provides abstractions and integrations for building applications efficiently, agnostic to the underlying model. Its components include modular abstractions, chains for specific use cases, and easy customization options.
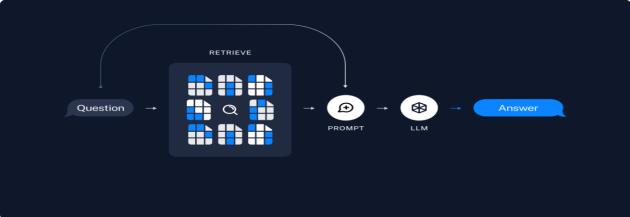
5. System Layers

• Reading & Processing PDF Files: Uses LangChain PyPDFDirectoryLoader to process PDFs.

• Document Chunking: Utilizes RecursiveCharacterTextSplitter to maintain semantic structure.

• Generating Embeddings: Employs OpenAIEmbeddings for text vectorization.

• Store Embeddings In ChromaDB: Uses CacheBackedEmbeddings to store vectors.

• Retrievers: Retrieves documents based on unstructured queries using VectorStoreRetriever.

• Re-Ranking with a Cross Encoder: Improves relevance using HuggingFaceCrossEncoder (BAAI/bge-reranker-base).

• Chains: Combines multiple components for a coherent application using prompts from LangChain Hub.

6. System Architecture

Refer to the project documentation for architecture diagrams.

7. Prerequisites

• Python 3.7+

• LangChain 0.3.13

• Add your OpenAI API key in a file named "OpenAI_API_Key.txt" to access OpenAI services.

8. Running

Steps to run the project:

• Clone the GitHub repository using: git clone
https://github.com/AkashNande/Semantic-Spotter-Project.git

• Open the Jupyter notebook and execute all cells to run the system.