Symantic Spotter using LlamaIndex

Project Objective :
Develop a project in the insurance sector that mirrors the one presented in the 'Retrieval Augmented Generation' session. The objective is to create a robust generative search system that can accurately and efficiently respond to inquiries based on various policy documents. You can utilize either LangChain or LlamaIndex to construct the generative search application.

Solution Strategy :

Develop a solution using LlamaIndex that meets the following requirements:

- **Knowledge Base Access**: Users should be able to retrieve responses from an insurance policy knowledge base.
- **Accurate Query Handling**: When a user performs a query, the system must accurately respond to the query based on the knowledge base.

Goal :
Successfully addressing the above two requirements will ensure a high level of accuracy for the overall model.

Data Used
Various HDFC insurance policy documents are stored in a single folder.

Tools Used
LlamaIndex and ChatGPT have been utilized for their powerful query engine, efficient data processing capabilities through data loaders and directory readers, and for enabling quicker implementation with minimal code. Additionally, disk caching has been employed to enhance performance.

Why LlamaIndex?
LlamaIndex is an innovative data framework specifically designed to support the development of LLM-based RAG (Retrieval-Augmented Generation) applications. It provides a robust platform that allows developers to seamlessly integrate various data sources with large language models.

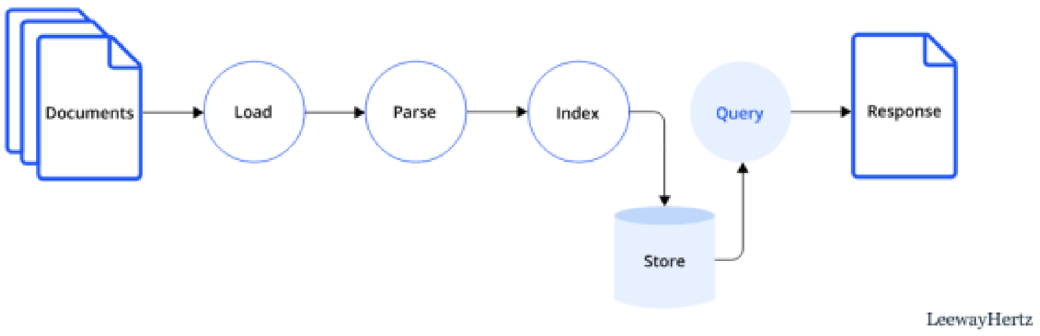
LlamaIndex supports multiple file formats, including PDFs and PowerPoints, as well as applications like Notion and Slack, and databases such as Postgres and MongoDB.

The framework offers a variety of connectors to facilitate data ingestion, ensuring smooth interactions with LLMs. Additionally, LlamaIndex features an efficient data retrieval and query interface.

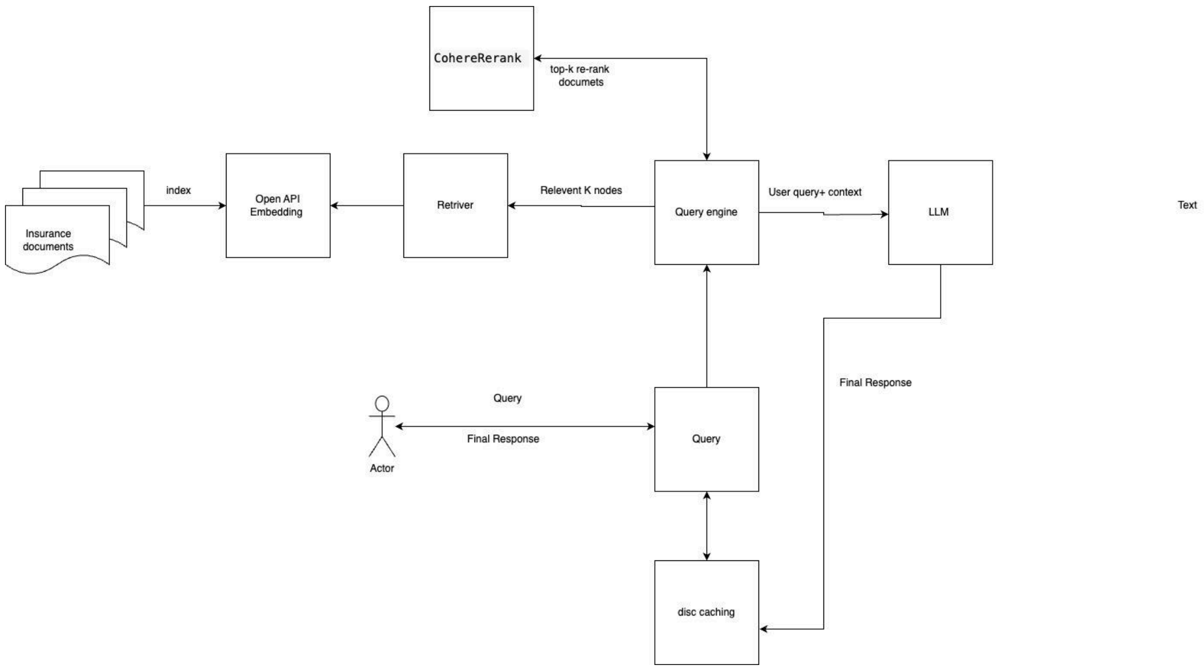
With LlamaIndex, developers can input any LLM prompt and receive outputs that are rich in context and knowledge augmentation.

Key Features of LlamaIndex:

- **Data Connectors**: Allows ingestion from a wide range of data sources and formats.
- **Data Synthesis**: Capable of synthesizing information from multiple documents or diverse data sources.
- **Extensive Integrations**: Offers numerous integrations with vector stores, ChatGPT plugins, tracing tools, LangChain, and more.



High Level Design of Semantic Spotter :



Architecture Descriptions:

1. **Documents**: A collection of HDFC insurance documents stored in a single folder.
2. **OpenAPI Embedding**: OpenAPI embeddings are used as the vector database for indexing the insurance documents.
3. **Query Engine**: The Query Engine module of LlamaIndex is employed for performing semantic searches. It utilizes an internal retriever and Cohere Rerank to retrieve the top-k relevant nodes from the embeddings.
4. **LLM Integration**: The top-k documents, along with the user query, are passed to the ChatGPT LLM to generate an accurate response.
5. **Caching**: Caching is implemented to enhance read operations. Recent similar searches are stored in the cache, and user queries are initially served from there. If a query is not found in the cache, it is forwarded to the query engine, and then the LLM generates the response. The user query and the generated response are cached for future use based on TTL (time-to-live).
6. **Metadata**: Along with the response, the system returns document references and similarity scores to boost user confidence in the implemented RAG system.
7. **Cohere Rerank**: This component is used to rerank the queries based on their semantic scores, ensuring that the most relevant results are prioritized.

Generative Search Response from Insurance Documents

We have included custom query generative search results below.

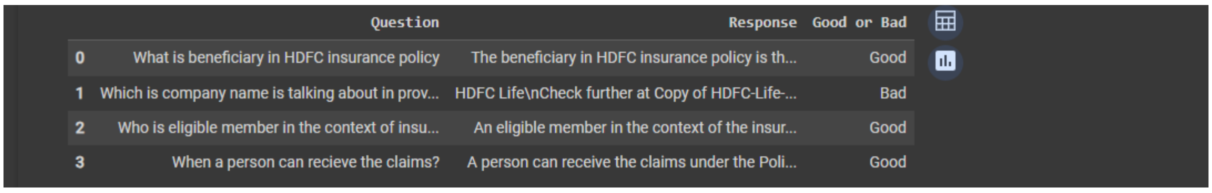
1. **Single Query Response**:
   The system processes a user query and retrieves relevant information from the insurance documents, providing a concise and contextually accurate response.
   Example Query: "What are the coverage options available in my HDFC health insurance policy?"
   Response:
   "The HDFC health insurance policy offers various coverage options, including hospitalization expenses, pre-and post-hospitalization expenses, day care procedures, and coverage for specific critical illnesses. Additional riders for maternity benefits and outpatient treatment can also be included in your policy."

Generative Search Response from Insurance Documents

We have included custom query generative search results below.

1. **Single Query Response**:
   The system processes a user query and retrieves relevant information from the insurance documents, providing a concise and contextually accurate response.

Example Query: "What are the coverage options available in my HDFC health insurance policy?"
Response:
"The HDFC health insurance policy offers various coverage options, including hospitalization expenses, pre-and post-hospitalization expenses, day care procedures, and coverage for specific critical illnesses. Additional riders for maternity benefits and outpatient treatment can also be included in your policy."



	Question	Response	Good or Bad
0	What is beneficiary in HDFC insurance policy	The beneficiary in HDFC insurance policy is th...	Good
1	Which is company name is talking about in prov...	HDFC Life\nCheck further at Copy of HDFC-Life-...	Bad
2	Who is eligible member in the context of insu...	An eligible member in the context of the insur...	Good
3	When a person can recieve the claims?	A person can receive the claims under the Poli...	Good

Challenges Faced:

We attempted to implement GPTCache for our caching system, but due to compatibility issues, we were unable to integrate it successfully.

Alternative Solution:

As an alternative to GPTCache, we are using disk caching.

Future Work:

1. **Feedback Mechanism**: We can enhance the solution by incorporating a feedback mechanism that embeds user responses along with feedback into the vector database.
2. **Integration of RAGAS/DeepEval Framework**: We can also integrate the RAGAS/DeepEval framework to further improve the semantic search capabilities and the quality of generated responses.