**PROJECT REPORT**

**HELPMATE AI PROJECT**

**Project Report**

**Title: Mr. HelpMate AI semantic search for Insurance Documents**

**Prepared By: Shruti, Sharmishtha, and Sheza**

**Date: 1-October-2024**

**Table of Content**

**1.** [**Problem Statement 2**](file:///C:\Users\sharm\Downloads\Project%20Report%20Mr.%20HelpMate%20AI.docx#_Toc162950643)

[**2. Objectives 2**](file:///C:\Users\sharm\Downloads\Project%20Report%20Mr.%20HelpMate%20AI.docx#_Toc162950644)

**3. Overview 3**

[**3. Scope and Requirements 3**](file:///C:\Users\sharm\Downloads\Project%20Report%20Mr.%20HelpMate%20AI.docx#_Toc162950645)

[**4. System Design 4**](file:///C:\Users\sharm\Downloads\Project%20Report%20Mr.%20HelpMate%20AI.docx#_Toc162950646)

**5. Challenges 4**

[**5. Learnings and handeling challenges 5**](file:///C:\Users\sharm\Downloads\Project%20Report%20Mr.%20HelpMate%20AI.docx#_Toc162950647)

[**6. Conclusion 5**](file:///C:\Users\sharm\Downloads\Project%20Report%20Mr.%20HelpMate%20AI.docx#_Toc162950648)

**Retrieval Augmented Generation with LlamaIndex**

1. **Problem Statement:**

This is a project in the insurance domain. In complex insurance documents locating specific information is a challenge for users like agents and policyholders. To address such issues, we are building a robust generative semantic search system with powerful language model that can deliver effective and accurate answers to user queries from various insurance policy documents.

1. **Objectives:**
   1. Design a system to accurately and efficiently extract text from insurance PDFs.
   2. Implement semantic search to enhance the relevance of results compared to keyword-based methods.
   3. Build a robust generative search system capable of effectively and accurately answering questions from various insurance policy documents.
   4. Use LangChain or LlamaIndex to build the generative search application.
2. **Overview:**

We will start with PDF processing and text extraction and then use the data framework, Llhamaindex supports ingesting data from various sources like text files, web pages, PDFs, and even structured data like CSV files.

After data ingestion, it builds an index that allows for efficient retrieval and querying of the data. Here a query engine is provided that enables users to ask natural language questions and receive relevant answers from the indexed data.

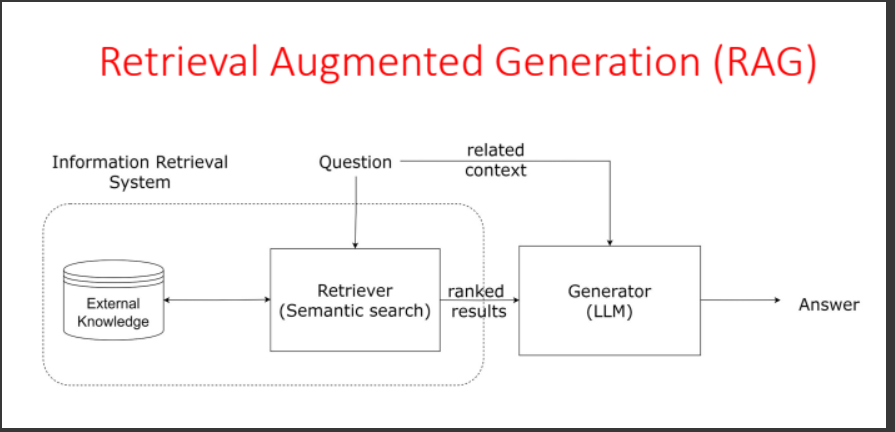
Llhamaindex is the ideal framework as it offers customizable retrieval strategies, such as vector similarity search, keyword search, and more, to find the most relevant data for a given query and streamlines information retrieval.

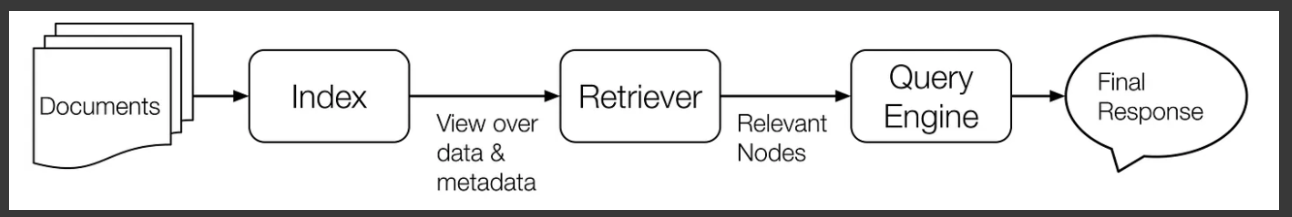
1. **Scope and Requirements**

* Features
  + PDF Preprocessing: (pdfplumber library)
  + Semantic Search: (ChromaDB, OpenAI embeddings)
  + Question Answering: (OpenAI language models)
  + Potential Additional Features: Table extraction and reformatting, answer summarization
* Technology Stack
  + Jupyter Notebook (Python)
  + pdfplumber: PDF parsing
  + OpenAI: Text embeddings, language model access
  + ChromaDB: Semantic search database with embedding storage
  + Sentence Transformers: Cross-encoder for result re-ranking (optional)
  + Other Libraries: tiktoken for tokenization, pandas for JSON parsing, pandasfor DataFrame manipulations

1. **System Design**

RAG System Architecture





1. **Challenges:**
2. Handling complex PDF structures
3. Language and terminology specific to insurance domain
4. API cost
5. Answer granularity
6. **Learnings and handling challenges:**
7. PDF analysis and using Simple DirectoryReader to extract all the PDFs from the folder directly
8. For language model accuracy fine-tune the language model on a terminology of insurance-related texts to improve its understanding and generation capabilities
9. A mix of APIs and local models can be cost effective
10. Hybrid retrieval to enhance the query type.
11. **Conclusion**
    1. Mr. HelpMate AI successfully demonstrates the potential of semantic search and AI-powered question answering to transform how users navigate insurance documents.
    2. The project highlighted the value of PDF preprocessing, embedding-based search, and language model fine-tuning.