HELP MATE AI

SUBTITLE : -
AI-POWERED INTELLIGENT SEARCH ASSISTANT

AUTHOR : -
ARNAB BERA (ML C63)

Table of Contents

Introduction

- **Introduction**

Generative Search Help Mate AI is a system designed to perform intelligent document searches using semantic search and re-ranking techniques. It processes PDF documents, retrieves relevant information, and generates responses based on the search results.

- **Problem Statement**

Traditional keyword-based searches often fail to retrieve accurate information from complex insurance policy documents due to ambiguous terms and lack of contextual understanding. This project aims to build a Retrieval-Augmented Generation (RAG) based generative search system that enhances search accuracy by:

- Using efficient text chunking for better document processing.
- Leveraging semantic search and re-ranking for relevant results.
- Generating context-aware answers using a robust AI model.

This system will provide precise, efficient, and user-friendly access to policy information, overcoming the limitations of conventional search methods.

Approach

Objectives

- To develop an AI-powered search system that retrieves contextually relevant information.
- To integrate semantic search and re-ranking techniques for better search results.
- To provide an efficient and user-friendly document search experience.

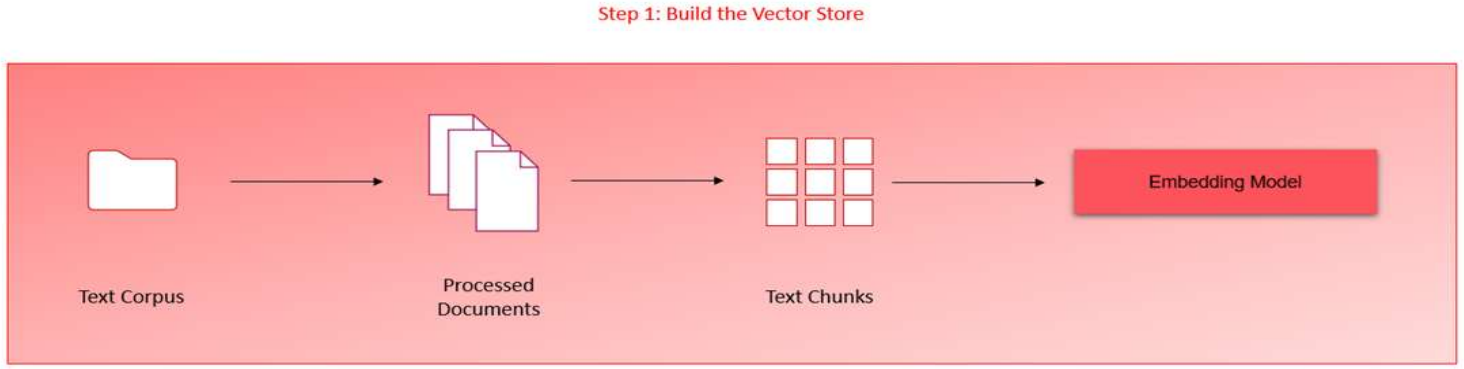
Approach

- Extract and preprocess text from PDF documents.
- Generate embeddings for document chunks using transformer-based models.
- Perform semantic search with caching for efficiency.
- Re-rank search results to improve relevance.
- Generate responses based on retrieved information.

System Architecture

Embedding Layer:

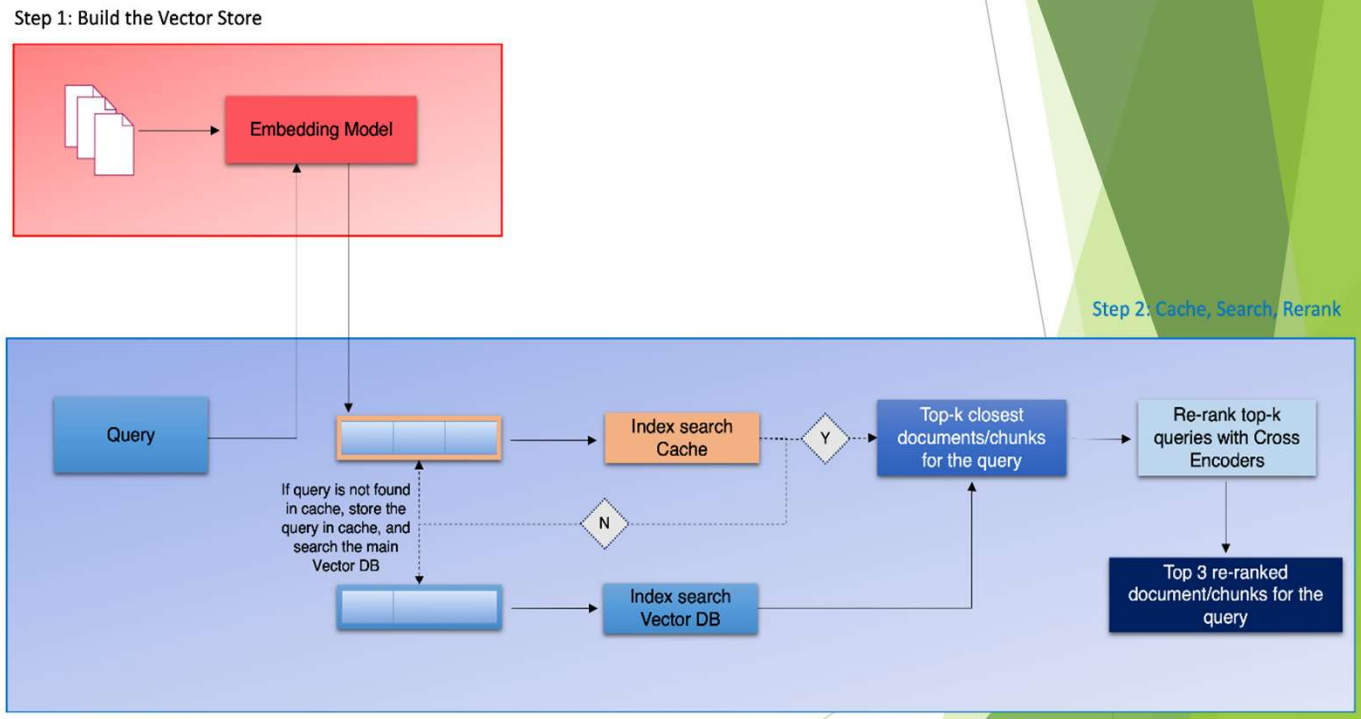
- Document Extraction: Use “pdfplumber” to extract text and tables from PDFs.
- Data Structuring: Store extracted data in a Data-frame.
- Embedding Generation: Convert data to vector representations using OpenAI’s text-embedding-ada-002 or Sentence Transformers.
- Storage: Store embeddings in Chroma DB.



System Architecture

Search Layer:

- Query Processing: Convert user queries to vector embeddings.
- Semantic Search: Perform similarity search in Chroma DB.
- Cache: Retrieve from cache if query exists.
- Re-ranking: Use cross-encoding models for result refinement.



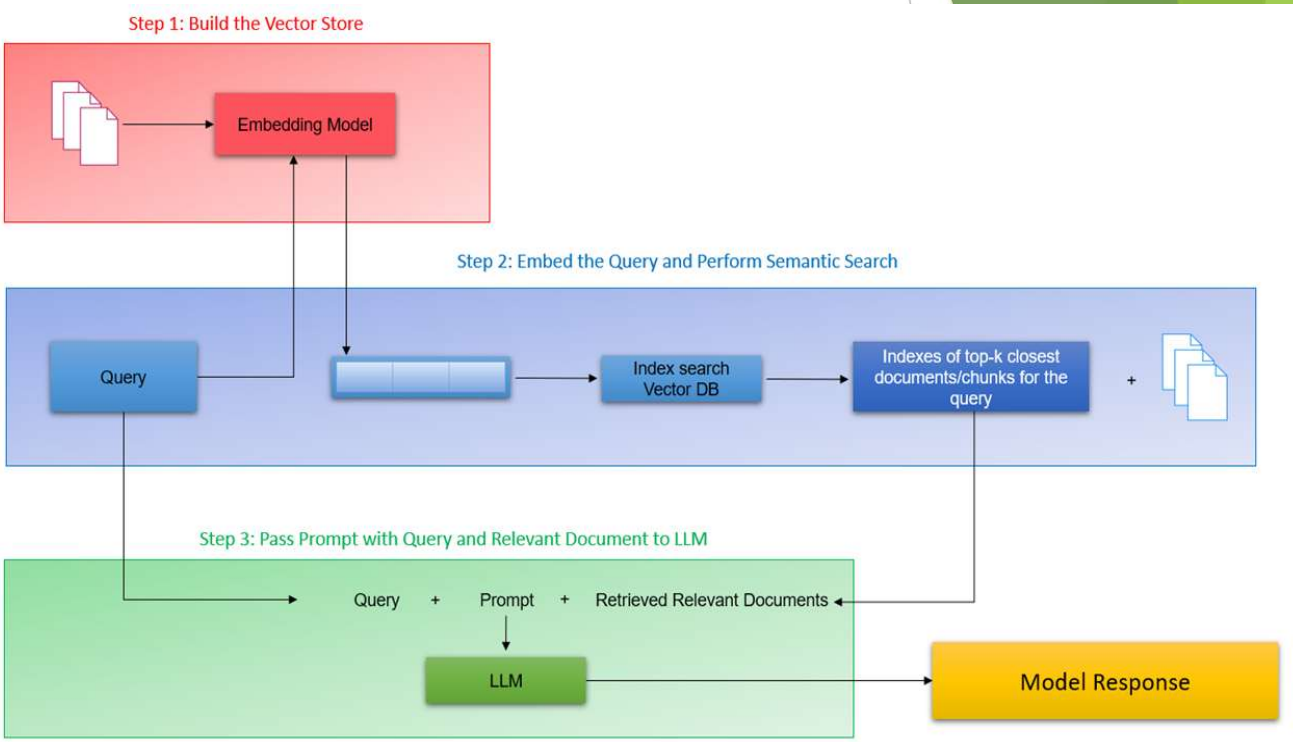
System Architecture

Generation Layer:

Prompt Engineering: Design prompts using query and relevant document chunks.

Cache Storage:

- Efficient Cache: Store queries and results in ChromaDB for faster retrieval.
- Threshold: Use semantic similarity threshold (e.g., 0.2) for cache management.



Key Functions

- **`semantic_search(input_query)` :**

Performs a semantic search for the given input query and returns the search results.

- **`rerank_scores(input_query, search_df)` :**

Re-ranks the search results based on their relevance to the input query.

- **`search_results_RAG(input_query)`:**

Combines semantic search and re-ranking to generate the final response for the input query.

- **`generative_search(query, top_3_RAG)`:**

Generate a response using GPT-3.5's ChatCompletion based on the user query and retrieved information.

Results & Demonstration

- **Sample Query**

“ What are the default benefits and provisions of the Group Policy? “

```
# Generate the response - For Query 1
top_3_RAG,response=search_results_RAG(query1)
```

```
Query:
 What are the default benefits and provisions of the Group Policy?
 ==================================================
 Result:
The default benefits and provisions of the Group Policy typically include various coverage options and administrative details. Based on the relevant documents, here are some key points regarding the default benefits and provisions:

1. **Discretion of the Principal**: The Principal has complete discretion to consider various factors when administering the policy, which may affect the benefits provided.

2. **Policy Administration**: The policy outlines specific administrative procedures that must be followed, ensuring that all parties are aware of their responsibilities and the processes involved.

3. **Statutory Requirements**: There are forms and documentation that must be completed to comply with statutory requirements, which are essential for the validity of the policy.

Unfortunately, the specific details regarding the benefits and provisions were not explicitly listed in the provided documents. For more comprehensive information, you may want to refer to the sections on policy administration and benefits in the cited pages.

| Key Points                        | Description                                                                        |
|-----------------------------------|------------------------------------------------------------------------------------|
| Discretion of the Principal       | The Principal has complete discretion in policy administration.                    |
| Policy Administration             | Specific procedures for administration are outlined in the policy.                 |
| Statutory Requirements            | Compliance with statutory requirements is necessary for policy validity.           |

For further details, please refer to the following pages of the policy document:

- **Policy Name**: Principal Group Policy, **Page No.**: 19
- **Policy Name**: Principal Group Policy, **Page No.**: 18
- **Policy Name**: Principal Group Policy, **Page No.**: 16

These sections may provide additional insights into the default benefits and provisions of the Group Policy.
```

```
# Check Top 3 results from semantic search
print("Query: ","\n",query1,"\n","="*50,"\n", "Result: ")
display(top_3_RAG)
```

```
Query:
 What are the default benefits and provisions of the Group Policy?
 ==================================================
 Result:
```

	Documents	Metadatas
0	T he Principal has complete discretion to cons...	{'Page_No.': 'Page 19', 'Policy_Name': 'Princi...
2	c . a copy of the form which contains the stat...	{'Page_No.': 'Page 18', 'Policy_Name': 'Princi...
1	PART II - POLICY ADMINISTRATION Section A - Co...	{'Page_No.': 'Page 16', 'Policy_Name': 'Princi...

Conclusions

- Generative Search Help Mate AI enhances search accuracy with semantic search and re-ranking techniques.
- Delivered a scalable and efficient solution for document retrieval and information extraction. Successfully implemented a semantic search system with the RAG pipeline and cache layer.
- Enables efficient retrieval of relevant information from large document repositories.
- Successfully implemented a semantic search system with the RAG pipeline and cache layer.

Challenges

Challenges:

- Performance Scaling: Address concerns about system performance with an increased number of documents or users by implementing vector databases and scaling up compute units.
- Cache Storage: Optimize the cache collection to efficiently store and retrieve queries and results.

Lessons Learned:

- Efficient Document Processing: Processing PDFs efficiently is crucial; libraries like “pdfplumber” and suitable data structures for storage play a vital role.
- Semantic Search Optimization: Fine-tune semantic search parameters and thresholds for optimal results.
- Cache Management: Implement an effective cache management strategy to balance storage and retrieval efficiency.

Acknowledgements

- The project references course materials from upGrad's curriculum.
- The project references presentations in upGrad’s recorded module given by Aditya Bhattacharya.
- The project references presentations in upGrad’s recorded module given by Akshay Ginodia.
- The project references insights and inferences from presentations in upGrad’s doubt clear session given by Shridhar Galande.
- The project references presentations in upGrad's live class given by Sheshanth AS.

THANK YOU !!!

CONTACT ME : -

- LinekdIn - https://www.linkedin.com/in/arnabbera-tech/
- Git Hub Repo - https://github.com/arnabberawork/Help-Mate-AI

UpGrad