Mr.HelpMateAI – RAG System for Insurance Policy Document

Abstract

This report presents the design and implementation of Mr.HelpMateAI, a Retrieval-Augmented Generation (RAG) system tailored for querying insurance policy documents. The system addresses the challenge of extracting meaningful insights from complex, unstructured PDF files by combining document parsing, semantic chunking, cache search and vector embeddings. Using pdfplumber, the system processes mixed-content documents and structures them into searchable chunks. These chunks are embedded using transformer-based models and stored in a ChromaDB vector database for efficient retrieval. A semantic search mechanism retrieves relevant chunks based on user queries, which are then reranked using a cross-encoder to improve relevance. Finally, GPT-3.5 generates user-facing answers with citations from the source document. The report details the architecture, implementation strategies, challenges encountered, and recommendations for future enhancements.

Introduction

This report presents a comprehensive overview of the implementation of a Retrieval-Augmented Generation (RAG) system designed to process and intelligently query insurance policy documents. The system is built to enable semantic search and natural language question answering over complex PDF-based insurance documents. The project leverages modern NLP techniques, vector embeddings, and generative models to deliver accurate, context-aware responses with citations.

Project Objectives

The primary goals of this project are as follows:

- To build a scalable and intelligent RAG pipeline for insurance policy documents.
- To extract and structure content from PDFs, including both text and tabular data.
- To generate semantic embeddings for document chunks and store them in a vector database.
- To implement a semantic search mechanism that retrieves relevant chunks based on user queries.
- To re-rank retrieved results using a cross-encoder for improved relevance.
- To generate final answers using GPT-3.5 model with citations from the source document.

Data Sources and Preprocessing

Source Document

- **File**: Principal-Sample-Life-Insurance-Policy.pdf
- **Content**: 64 pages covering various aspects of life insurance including eligibility, benefits, termination, continuation, reinstatement, and claims.
- **Format**: PDF with mixed content (text and tables).

Preprocessing Pipeline

- **Library Used**: pdfplumber
- **Steps**:
    - Open and iterate through each page.
    - Extract text and tables separately using bounding box logic.
    - Cluster content to preserve original layout.
    - Filter out pages with fewer than 10 words.
    - Store structured data in a Pandas DataFrame with metadata (Page No., Policy Name).

System Architecture and Design

Chunking Strategy

- **Method**: Fixed-size chunking (500 words per chunk).
- **Purpose**: To ensure manageable input sizes for embedding and retrieval.
- **Metadata**: Each chunk is tagged with Page No. and Chunk No. for traceability.

Embedding Generation

- **Model**: all-MiniLM-L6-v2 from sentence_transformers
- **Functionality**:
    - Converts each chunk into a dense vector representation.
    - Embeddings are stored in the DataFrame for later retrieval.

Vector Store Integration

- **Tool**: ChromaDB
- **Collections**:
  - RAG_on_Insurance: Stores chunk embeddings and metadata.
  - Insurance_Cache: Stores query embeddings and previously retrieved results.
- **Storage**: Embeddings, chunk text, and metadata are indexed for fast retrieval.

Semantic Search and Reranking

- **Initial Retrieval**: Based on cosine similarity of embeddings.
- **Reranking**:
  - Model: cross-encoder/ms-marco-MiniLM-L-6-v2
  - Scores query-response pairs for relevance.
  - Top results are selected for final answer generation.

Generation Layer

- **Model**: GPT-3.5
- **Prompt Engineering**:
  - Includes user query, top 3 retrieved chunks, and metadata.
  - Generates a direct answer with citations.
- **Output**: Human-readable response with policy name and page number references.

Key Implementation Methods

- **PDF Extraction Function: extract_text_from_pdf()**
  - Extracts text and tables from each page using pdfplumber.
  - Separates table content using bounding box logic.
  - Clusters content to preserve original layout and chronology.

- **Chunking Function: split_text_into_chunks()**
  - Splits long page texts into fixed-size chunks (e.g., 500 words).
  - Ensures each chunk is manageable for embedding and retrieval.
  - Returns a list of chunked text segments.
- **Chunk Processing Function: process_page()**
  - Applies chunking to the page content.
  - Retrieves page text and adds Chunk_No. to metadata for each chunk.
- **Embedding Generation Function: generate_embeddings_on_df()**
  - Uses SentenceTransformer to encode each chunk into a vector.
  - Stores embeddings in the DataFrame under the Embeddings column.
  - Enables semantic similarity search via vector comparison.
- **Reranking Function: reranking()**
  - Pairs each retrieved chunk with the user query.
  - Scores relevance using a cross-encoder model.
  - Sorts and selects top results based on reranked scores.
- **Search Function**: **search()**
  - Checks cache first.
  - Falls back to main collection if cache miss.
  - Stores new queries in cache for future reuse.
- **Answer Generation Function: generate_response()**
  - Constructs a prompt with query and top retrieved chunks.
  - Sends prompt to GPT-3.5 for natural language response.
  - Returns a formatted answer with citations from the source document.

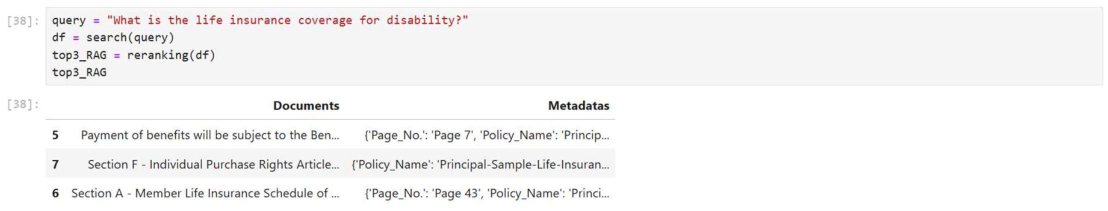
Challenges Faced

Challenge	Resolution
Mixed content in PDF	Used bounding box logic to separate tables from text.
Empty or irrelevant pages	Filtered using word count threshold.
Embedding latency	Batched encoding to improve performance.
Telemetry errors in ChromaDB	Suppressed warnings and continued execution.
Query relevance	Improved using cross-encoder reranking.

Sample Queries and Results

Queries from Search Layer

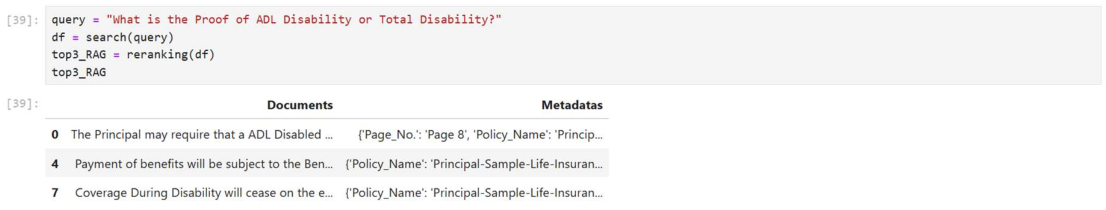
Q1. What is the life insurance coverage for disability?



```
[38]: query = "What is the life insurance coverage for disability?"
      df = search(query)
      top3_RAG = reranking(df)
      top3_RAG
```

```
[38]:                     Documents                                      Metadatas
      5  Payment of benefits will be subject to the Ben...   {'Page_No.': 'Page 7', 'Policy_Name': 'Princip...
      7     Section F - Individual Purchase Rights Article...  {'Policy_Name': 'Principal-Sample-Life-Insuran...
      6  Section A - Member Life Insurance Schedule of ...   {'Page_No.': 'Page 43', 'Policy_Name': 'Princi...
```

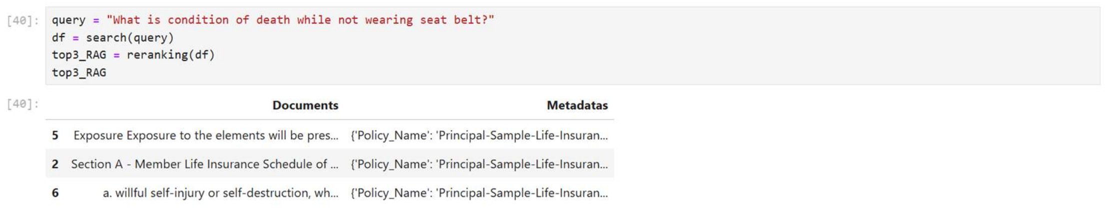
Q2. What is the Proof of ADL Disability or Total Disability?



```
[39]: query = "What is the Proof of ADL Disability or Total Disability?"
      df = search(query)
      top3_RAG = reranking(df)
      top3_RAG
```

```
[39]:                     Documents                                      Metadatas
      0  The Principal may require that a ADL Disabled ...   {'Page_No.': 'Page 8', 'Policy_Name': 'Princip...
      4  Payment of benefits will be subject to the Ben...   {'Policy_Name': 'Principal-Sample-Life-Insuran...
      7  Coverage During Disability will cease on the e...   {'Policy_Name': 'Principal-Sample-Life-Insuran...
```

Q3. What is condition of death while not wearing seat belt?

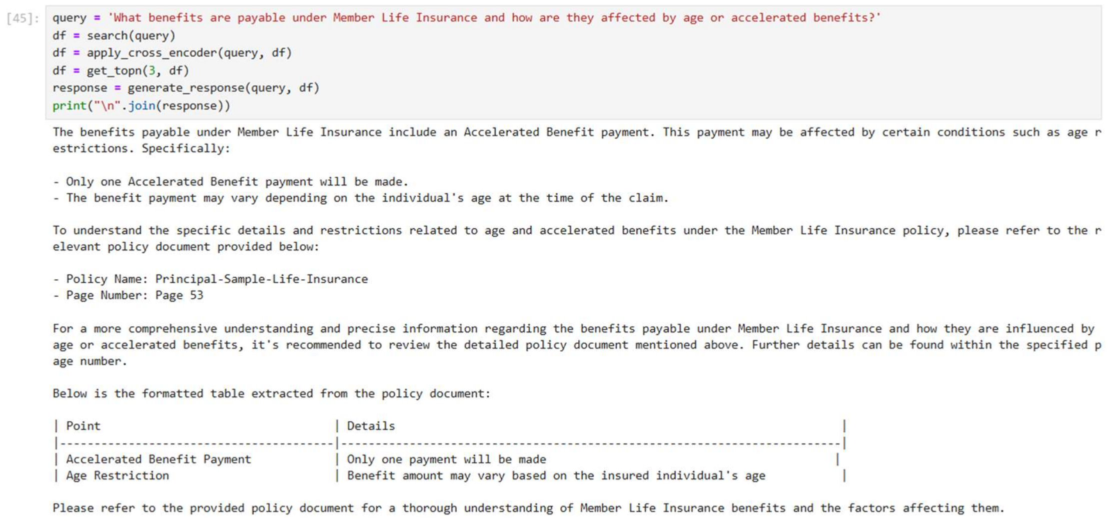


```
[40]: query = "What is condition of death while not wearing seat belt?"
      df = search(query)
      top3_RAG = reranking(df)
      top3_RAG
```

```
[40]:                     Documents                                      Metadatas
      5  Exposure Exposure to the elements will be pres...   {'Policy_Name': 'Principal-Sample-Life-Insuran...
      2  Section A - Member Life Insurance Schedule of ...   {'Policy_Name': 'Principal-Sample-Life-Insuran...
      6         a. willful self-injury or self-destruction, wh...   {'Policy_Name': 'Principal-Sample-Life-Insuran...
```

Queries from Generation Layer

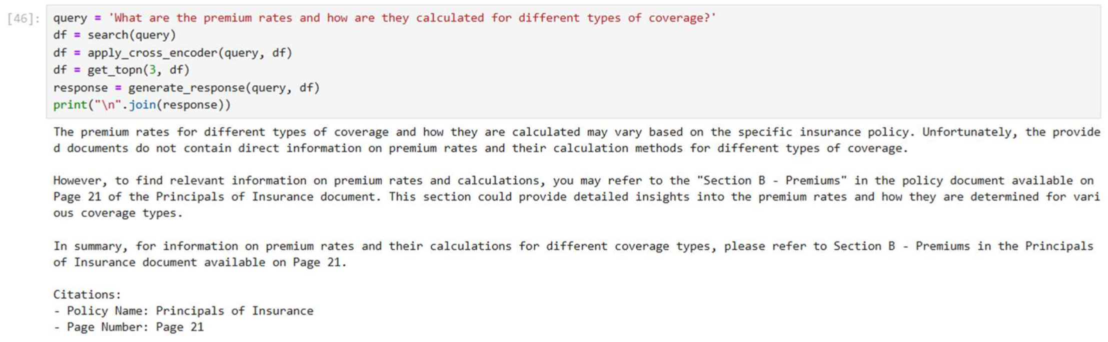
Q1. What benefits are payable under Member Life Insurance and how are they affected by age or accelerated benefits?



```
[45]: query = 'What benefits are payable under Member Life Insurance and how are they affected by age or accelerated benefits?'
      df = search(query)
      df = apply_cross_encoder(query, df)
      df = get_topn(3, df)
      response = generate_response(query, df)
      print("\n".join(response))
```

```
The benefits payable under Member Life Insurance include an Accelerated Benefit payment. This payment may be affected by certain conditions such as age r
estrictions. Specifically:

- Only one Accelerated Benefit payment will be made.
- The benefit payment may vary depending on the individual's age at the time of the claim.

To understand the specific details and restrictions related to age and accelerated benefits under the Member Life Insurance policy, please refer to the r
elevant policy document provided below:

- Policy Name: Principal-Sample-Life-Insurance
- Page Number: Page 53

For a more comprehensive understanding and precise information regarding the benefits payable under Member Life Insurance and how they are influenced by
age or accelerated benefits, it's recommended to review the detailed policy document mentioned above. Further details can be found within the specified p
age number.

Below is the formatted table extracted from the policy document:

| Point                                  | Details                                                                 |
|----------------------------------------|-------------------------------------------------------------------------|
| Accelerated Benefit Payment            | Only one payment will be made                                          |
| Age Restriction                        | Benefit amount may vary based on the insured individual's age           |

Please refer to the provided policy document for a thorough understanding of Member Life Insurance benefits and the factors affecting them.
```

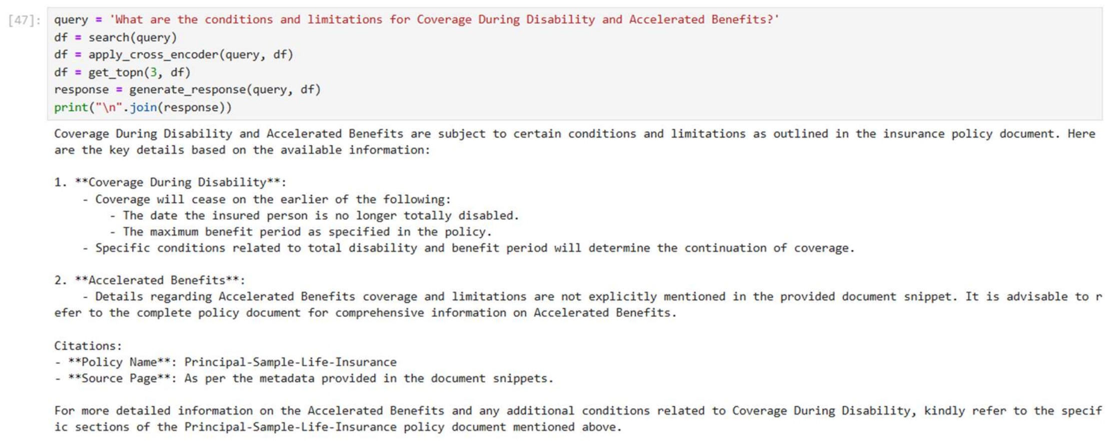
Q2. What are the premium rates and how are they calculated for different types of coverage?



```
[46]: query = 'What are the premium rates and how are they calculated for different types of coverage?'
      df = search(query)
      df = apply_cross_encoder(query, df)
      df = get_topn(3, df)
      response = generate_response(query, df)
      print("\n".join(response))
```

```
The premium rates for different types of coverage and how they are calculated may vary based on the specific insurance policy. Unfortunately, the provide
d documents do not contain direct information on premium rates and their calculation methods for different types of coverage.

However, to find relevant information on premium rates and calculations, you may refer to the "Section B - Premiums" in the policy document available on
Page 21 of the Principals of Insurance document. This section could provide detailed insights into the premium rates and how they are determined for vari
ous coverage types.

In summary, for information on premium rates and their calculations for different coverage types, please refer to Section B - Premiums in the Principals
of Insurance document available on Page 21.

Citations:
- Policy Name: Principals of Insurance
- Page Number: Page 21
```

Q3. What are the conditions and limitation for Coverage during Disability and Accelerated Benefits?



```
[47]: query = 'What are the conditions and limitations for Coverage During Disability and Accelerated Benefits?'
      df = search(query)
      df = apply_cross_encoder(query, df)
      df = get_topn(3, df)
      response = generate_response(query, df)
      print("\n".join(response))
```

```
Coverage During Disability and Accelerated Benefits are subject to certain conditions and limitations as outlined in the insurance policy document. Here
are the key details based on the available information:

1. **Coverage During Disability**:
    - Coverage will cease on the earlier of the following:
        - The date the insured person is no longer totally disabled.
        - The maximum benefit period as specified in the policy.
    - Specific conditions related to total disability and benefit period will determine the continuation of coverage.

2. **Accelerated Benefits**:
    - Details regarding Accelerated Benefits coverage and limitations are not explicitly mentioned in the provided document snippet. It is advisable to r
efer to the complete policy document for comprehensive information on Accelerated Benefits.

Citations:
- **Policy Name**: Principal-Sample-Life-Insurance
- **Source Page**: As per the metadata provided in the document snippets.

For more detailed information on the Accelerated Benefits and any additional conditions related to Coverage During Disability, kindly refer to the specif
ic sections of the Principal-Sample-Life-Insurance policy document mentioned above.
```

Recommendations and Future Enhancements

- **Multi-document support**: Extend pipeline to handle multiple PDFs.
- **UI Integration**: Build a front-end using Streamlit or Flask.
- **Advanced chunking**: Use semantic chunking based on headings and sections.
- **Feedback loop**: Incorporate user feedback to refine cache and reranking.
- **Model upgrade**: Integrate OpenAI's "text-embedding-ada-002" for better semantic coverage.

Conclusion

Mr.HelpMateAI successfully demonstrates a robust RAG system tailored for insurance policy documents. It combines effective preprocessing, semantic search, reranking, and generative answering to deliver accurate, user-friendly responses. The modular design ensures scalability and adaptability to other domains such as legal, healthcare, and finance.