



Semantic Spotter 2.0

Project Report

Syed Abdul Rahim
sarpsl@gmail.com

Contents

- Project Goals 2
- Data Sources 2
 - Document Source: 2
- Design Choices 2
 - Framework Choice: 2
 - Document Parsing: 2
 - Vector Storage: 2
 - Query Processing: 2
 - Integration with LLM: 3
- Challenges Faced..... 3
 - Complexity of PDF Documents:..... 3
 - Indexing Efficiency: 3
 - Query Accuracy: 3
- Flowchart 3
- Setup 4
- Usage 4
 - Import Libraries:..... 4
 - Run the Script: 4
 - Query the System:..... 4
- Sample Run:..... 4
 - Use Case:..... 4
 - Chat Conversation: 5
- Troubleshooting 6
 - PDF Parsing Issues:..... 6
 - Indexing Errors: 6

Project Goals

This project aims to create a document processing and query system using LlamaIndex with an HDFC Policy PDF as the source document. The goal is to build a system that can parse complex documents, extract relevant information, and provide user-friendly responses to queries. By leveraging LlamaIndex, we aim to efficiently index and query document contents, offering a scalable solution for handling policy documents and similar texts.

Data Sources

Document Source: HDFC-Life-Group-Term-Life-Policy PDF

Purpose: The HDFC Policy PDF serves as the primary data source for this project. The document contains detailed information about insurance policies, which will be parsed and indexed.

Format: PDF

Content: Policy terms, conditions, coverage details, and other relevant sections.

Design Choices

Framework Choice: LlamaIndex is chosen for its capability to handle document parsing, indexing, and querying efficiently. It integrates well with various vector storage solutions and provides a robust querying engine.

Document Parsing: The PDF is parsed into structured data using LlamaIndex's `SimpleNodeParser`. This choice allows us to handle complex documents with multiple sections and formats.

Vector Storage: For simplicity and to maintain local control, the vector storage is handled directly within the LlamaIndex framework without external databases.

Query Processing: The system uses LlamaIndex's querying capabilities to search through the indexed document and retrieve relevant information based on user queries.

Integration with LLM: While not required in the current implementation, integration with a Language Model like OpenAI's GPT could be considered for enhanced interaction and natural language understanding.

Challenges Faced

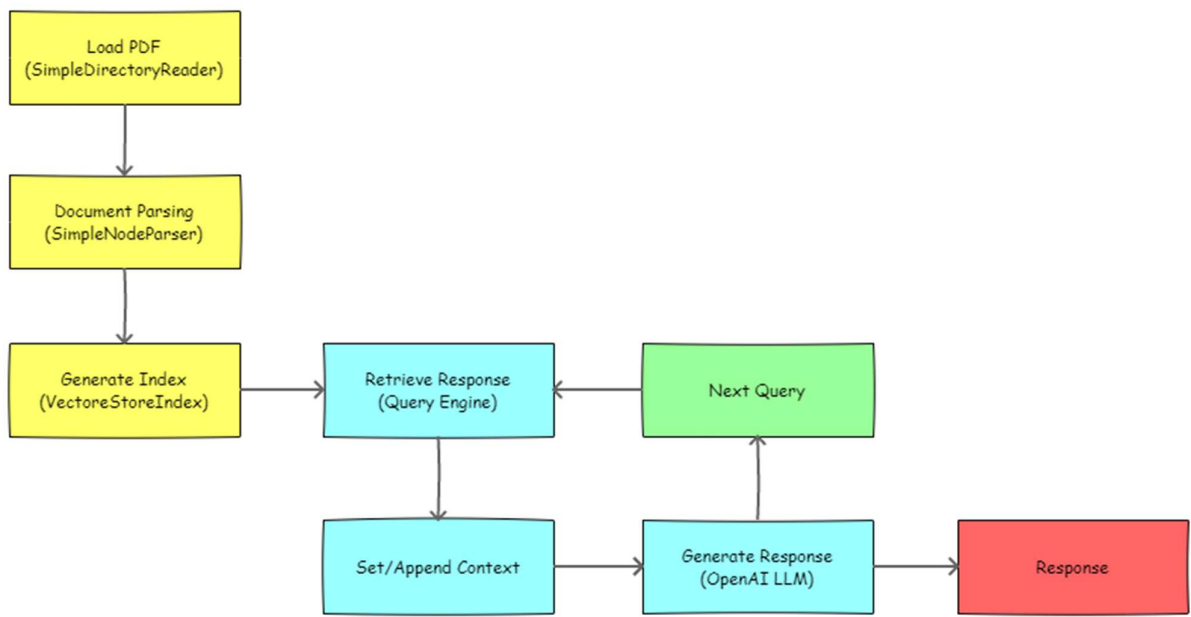
Complexity of PDF Documents: Parsing PDFs can be challenging due to the complex layout and varied formatting of documents. Ensuring that all relevant information is accurately extracted and indexed required careful handling.

Indexing Efficiency: Efficiently indexing large documents while maintaining performance and accuracy was a key challenge. Optimizations were needed to balance indexing speed and query performance.

Query Accuracy: Ensuring that the query results are accurate and relevant required fine-tuning of the parsing and indexing processes. Handling different types of queries and ensuring correct results involved iterative testing and adjustments.

Flowchart

A flowchart illustrating the system design and various layers of the project:



Setup

1. Place the HDFC Policy PDF file in the project directory.
2. Update the PDF path in the code if necessary.

Usage

Import Libraries:

If it's a fresh environment or the libraries aren't installed, then uncomment the first 'pip install' cell block and execute to get those libraries installed in your environment.

Run the Script:

It's a Jupyter notebook, feel free to open in your favourite editor, like VS Code or Google Collab and do Run All

Query the System:

Enter queries related to the HDFC policy to receive relevant information from the document.

Sample Run:

Use Case:

```
initial_query = "List all 'active' related definitions"
initial_answer = retrieve_and_generate(initial_query)
print("\nQuestion:", initial_query)
print("Answer:", initial_answer)

follow_up_query = "Is Active and Death related"
follow_up_answer = retrieve_and_generate(follow_up_query, False)
print("\nQuestion:", follow_up_query)
print("Answer:", follow_up_answer)
```

✓ 6.0s

Question: List all 'active' related definitions

Answer: Active Service shall mean employees employed with the Policyholder on a full time permanent basis and who are actively working on a day which is one of the Policyholder's scheduled work days and are performing in the customary manner all the regular duties of his employment with the Policyholder on a full-time basis that day either at one of the Policyholder's business establishments, or at some location to which the Policyholder's business requires him to travel. An employee will be considered in Active Service on a day, which is not one of the Policyholder's scheduled workdays only if he was performing in the customary manner all of the regular duties of his employment on the preceding scheduled workday. This is only applicable for

Question: Is Active and Death related

Answer: No, the term 'Active' is related to employment status and participation, not to death.

Chat Conversation:

```
tabnine: test | explain | document | ask
• def initialize_conv():
    print("1. new: for New Conversation")
    print("2. exit: once you are done")
    print("3. Your initial / follow-up question")
    new_conversation = True
    while True:
        user_input = input()
        if user_input.lower() == 'exit':
            print('\nExiting the program... bye')
            break
        elif user_input.lower() == 'new':
            print("\nStarting a new conversation")
            new_conversation = True
        else:
            question = user_input
            answer = retrieve_and_generate(question, new_conversation)
            print("\nQuestion:", question)
            print("Answer:", answer)
            new_conversation = False
```

✓ 0.0s

```
initialize_conv()
```

✓ 1m 0.2s

1. new: for New Conversation
2. exit: once you are done
3. Your initial / follow-up question

Question: Explain policy terms

Answer: The leaflet containing the simplified version of the provisions of Section 45 is enclosed as Annexure B for reference.

Question: List 3 terms

Answer: - Basic Policy Contract

- Fraud and Misstatement

- Eligibility Criteria

Starting a new conversation

Question: What is policy expiration?

Answer: Policy expiration occurs when the Premium due is not paid within the Grace Period, leading to the Policy lapsing and the Company ceasing to be liable to pay any Benefits under the lapsed Policy.

Question: What are the terms of policy expiration?

Answer: Policy expiration occurs when the Premium due is not paid within the Grace Period, leading to the Policy lapsing and the Company ceasing to be liable to pay any Benefits under the lapsed Policy.

Exiting the program... bye

Troubleshooting

PDF Parsing Issues: Ensure the PDF is correctly formatted and accessible.

Indexing Errors: Verify that all dependencies are correctly installed and configured.

[End of Document]