**Building the RAG - Report**

**Objective of Project:**

The objective of this project is to develop a Retrieval-Augmented Generation (RAG) system using LlamaIndex to process and analyze the LIC's New Jeevan Shanti policy document. The system will employ multiple embedding models to index the document, compare the performance of different large language models (LLMs) during the synthesis stage, and evaluate the generated responses based on faithfulness, correctness, and relevance.

**Workflow:**

**PDF Parsing**

* Use **LlamaParser** to extract structured and unstructured text from the LIC New Jeevan Shanti policy PDF.

**Vector Store Index Creation**

* Convert the parsed text into a **vector store index** for efficient retrieval of relevant information.

**Embedding Models**

* Apply **at least three different embedding models** to generate vector representations of the document content.
* Compare how different embeddings affect retrieval performance.

**Synthesis with LLMs**

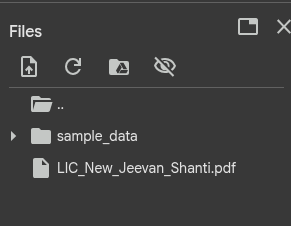
* Use **at least two different LLMs** to generate answers based on retrieved content.
* Compare the performance of the LLMs in terms of response quality.

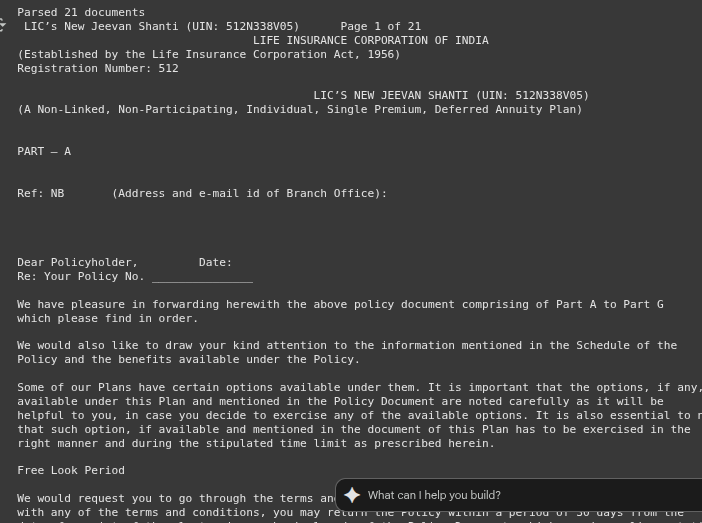
**Evaluation**

* Assess generated responses based on **faithfulness, correctness, and relevancy**.
* Analyze results to determine the best combination of embedding models and LLMs for the task.

**PDF Parsing:**

In this stage of the project, the LIC New Jeevan Shanti policy document was first downloaded from the official LIC website using Python’s requests library and saved locally as a PDF file. The downloaded document was then parsed using the PDFReader from the llama\_index.readers.file module to extract its textual content into structured documents. A total of *n* documents were obtained, with the first document inspected to ensure accurate parsing. This process establishes the foundation for subsequent steps in the RAG system, including vector indexing, embedding generation, and information retrieval, by providing clean and accessible textual data from the policy PDF.





**Vector Store Index Creation & Embedding Models:**

In this stage, three different embedding models—**MiniLM-MSMARCO**, **DistilBERT-MSMARCO**, and **Sentence-T5-Large**—were employed to generate vector representations of the parsed policy documents. Using **HuggingFaceEmbedding**, each model was configured to encode the textual content, capturing semantic meaning for efficient retrieval. Subsequently, **FAISS** was utilized to create separate vector store indexes for each embedding model, allowing similarity-based searches across the document corpus. Each index was built using the VectorStoreIndex from LlamaIndex, with the respective embedding model applied, ensuring that the documents were appropriately transformed into high-dimensional vector representations. This setup enables comparative analysis of different embeddings in terms of retrieval performance and lays the groundwork for the synthesis and evaluation stages of the RAG system.

**Why i choosed these three embedings models :**

**MiniLM-MSMARCO**

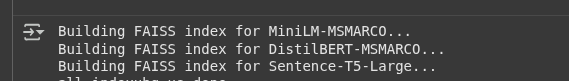
* Lightweight and efficient, suitable for **fast encoding** of large document collections.
* Optimized on the **MSMARCO dataset**, making it effective for **semantic search and retrieval** tasks.
* Balances **speed and accuracy**, ideal for building initial vector indexes.

**DistilBERT-MSMARCO**

* A **distilled version of BERT**, offering faster inference while retaining strong performance.
* Trained on **MSMARCO**, providing good **relevance for question-answering and retrieval**.
* Helps compare **trade-offs between model size, speed, and embedding quality**.

**Sentence-T5-Large**

* Large transformer model capable of producing **high-quality, dense embeddings**.
* Optimized for **sentence-level semantic similarity**, ensuring more nuanced understanding of text.
* Useful for **evaluating the impact of high-capacity models** on retrieval performance and RAG output.



**Synthesis with LLMs:**

**Flan-T5-base model**:

In this stage, a **query regarding the definitions of terms used in the policy** was executed on the RAG system. The Sentence-T5-Large embedding index was selected to retrieve the **top three most relevant document chunks** using semantic similarity. These retrieved chunks were then consolidated into a context string, which served as input for the **Flan-T5-base model** via Hugging Face’s text2text-generation pipeline. The model processed the context and generated an answer to the query, demonstrating the system’s ability to combine **vector-based retrieval with LLM-based synthesis**. This approach highlights the effectiveness of using high-quality embeddings for accurate information retrieval and the capability of the LLM to produce coherent and contextually relevant responses from multiple document segments.

**Parameters set up:**

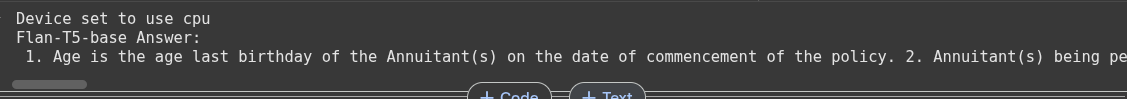
1. **Query**: "the definitions of terms/words used in the Policy?"
2. **Embedding Index Used**: Sentence-T5-Large for semantic retrieval.
3. **Retriever Settings**: Top **3 most similar document chunks** (similarity\_top\_k=3).
4. **Context Preparation**: Concatenated retrieved chunks into a single text input for the LLM.
5. **LLM Used**: Flan-T5-base via Hugging Face text2text-generation pipeline.
6. **Generation Parameters**:

max\_new\_tokens=512 (limit the output length)

truncation=True (truncate input if too long)

device=-1 (CPU inference; can change to GPU for speed).

**Response:**

****

**OPT-350M model:**

In this stage, the query "the definitions of terms/words used in the Policy?" was executed using the **DistilBERT-MSMARCO embedding index**. The **top three relevant document chunks** were retrieved and concatenated to form the context, which was truncated to a maximum of **512 tokens** to fit the model’s input constraints. The **OPT-350M model** was then used via Hugging Face’s text-generation pipeline to generate an answer. Key parameters included max\_new\_tokens=256 for output length, do\_sample=False for deterministic generation, and return\_full\_text=False to obtain only the generated answer. This stage demonstrates the system’s ability to combine a **different embedding model** with an alternative LLM, allowing comparison of response quality, relevance, and correctness across models in the RAG framework.

**Set up:**

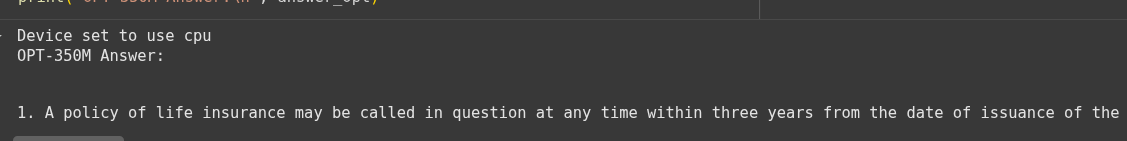
1. **Query**: "the definitions of terms/words used in the Policy?"
2. **Embedding Index Used**: DistilBERT-MSMARCO
3. **Retriever Settings**: Top **3 most similar document chunks** (similarity\_top\_k=3)
4. **Context Preparation**: Concatenated retrieved chunks and **truncated to 512 tokens**
5. **LLM Used**: OPT-350M via Hugging Face text-generation pipeline
6. **Generation Parameters**:

max\_new\_tokens=256 (limit the length of generated output)

do\_sample=False (deterministic output)

return\_full\_text=False (only return the generated answer)

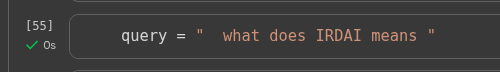
device=-1 (CPU inference; can use GPU for faster execution)



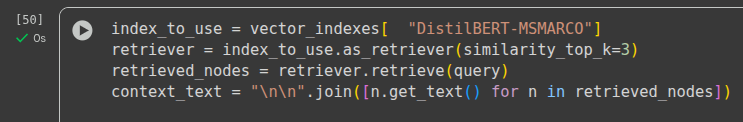
**Evaluation:**

In the evaluation stage, the quality of answers generated by different LLMs was assessed using **BERTScore**, which measures semantic similarity between the model-generated answers and the **reference (ground truth) answer**. The DistilBERT-MSMARCO embedding index was selected to retrieve relevant context for both **Flan-T5-base** and **OPT-350M** models. Precision, recall, and F1 scores were calculated for each LLM’s output, providing a quantitative assessment of **faithfulness, correctness, and relevance** of the generated responses. This approach enables a direct comparison of LLM performance when combined with a specific embedding model, helping to identify which model better captures the intended meaning of the policy document.

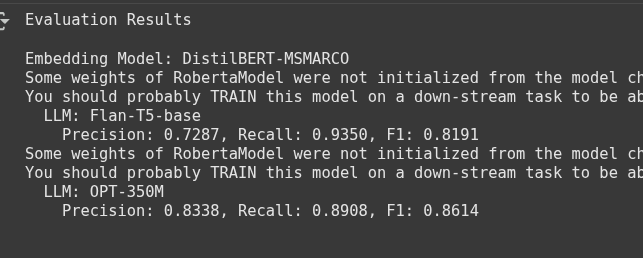
**Query:**

****

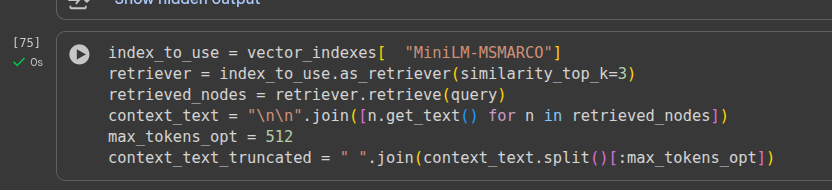
**Embedding mode: ("DistilBERT-MSMARCO")**

****

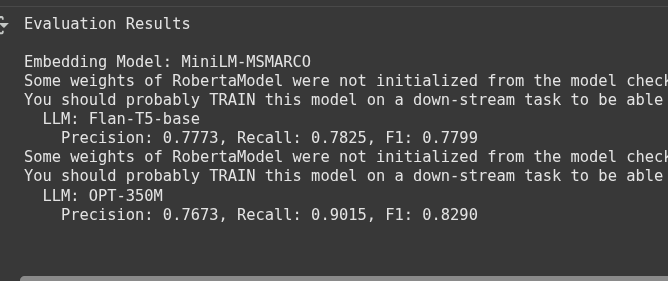
**Evaluation:**

****

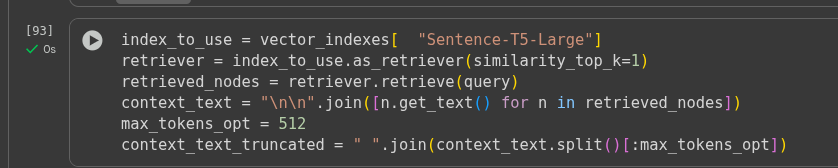
**Embedding mode: (MiniLM-MSMARCO)**

****

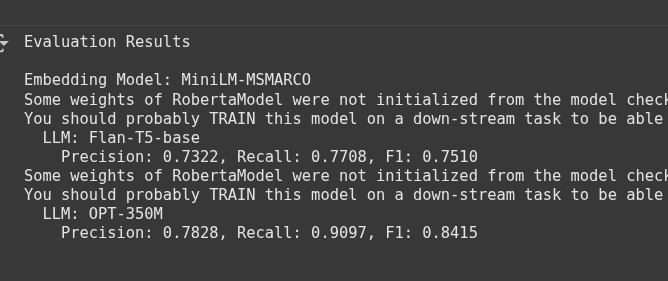
**Evaluation:**

****

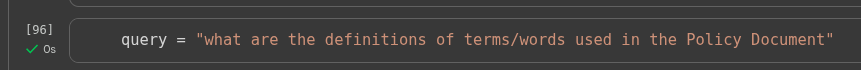
**Embedding mode:(Sentence-T5-Large)**

****

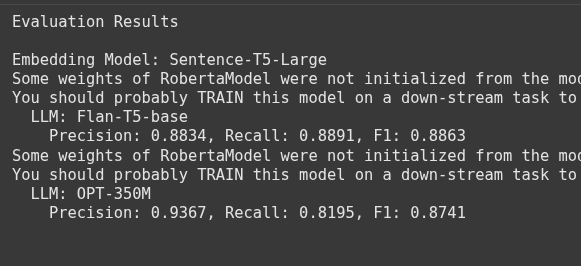
**Evaluation:**

****

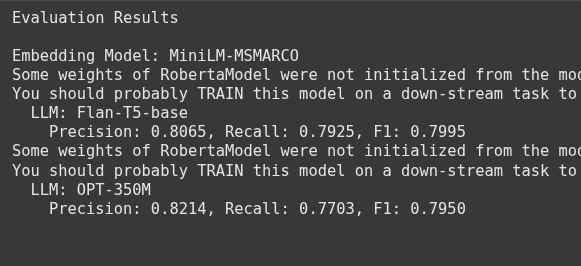
**Query:**

****

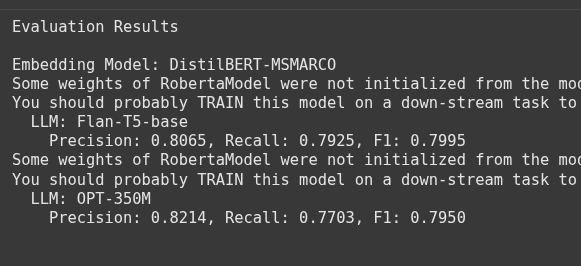
**Embedding model = “Sentence-T5-Large”**

****

**Embedding model = “MiniLM-MSMARCO”**

****

**Embedding model = “DistilBERT-MSMARCO”**

****