Name:-	**Shrutam S. Shah**
Course :-	**B.Tech in Computer Science and Engineering (pre-final year)**
University:-	**Nirma University**
Current CGPA:-	**8.13**
Personal E-mail Id:- **University E-mail Id:-**	**shrutam17@gmail.com** **21bce273@nirmauni.ac.in**

Task :-

Create a RAG-powered chatbot that can answer questions based on a PDF document.

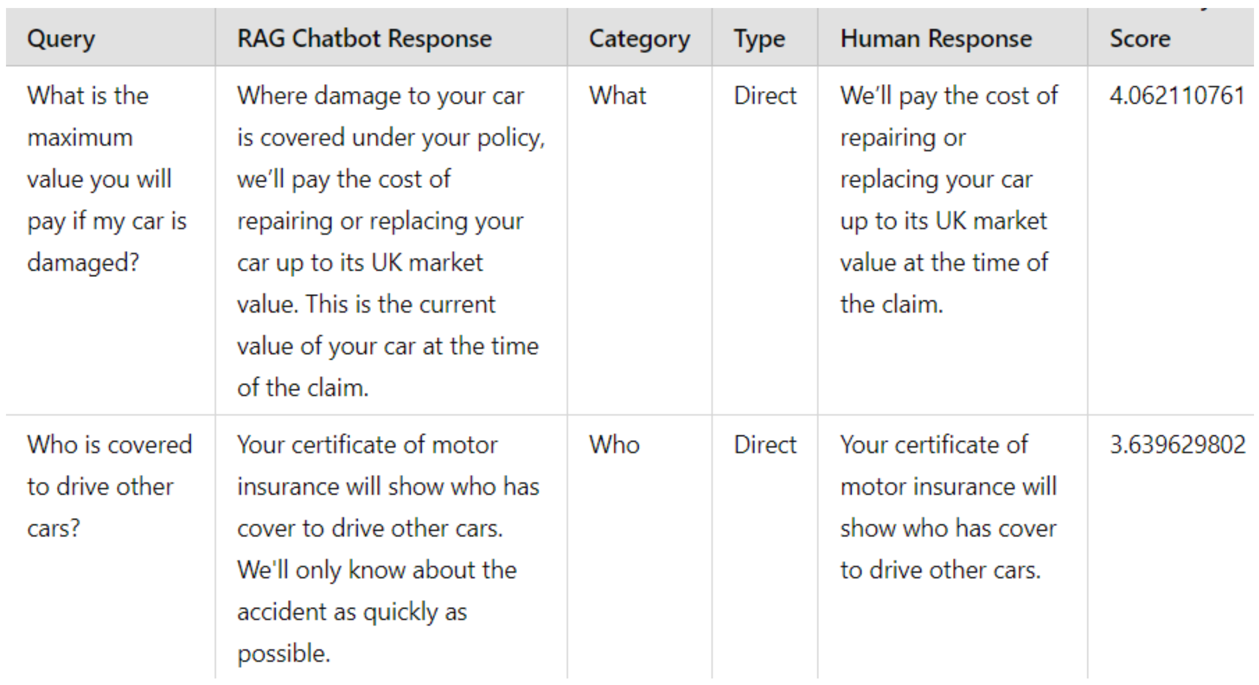
Introduction :-

Here I have designed a RAG (Retrieval-Augmented Generation) powered chatbot using the PDF document as the knowledge source. The chatbot answers the questions related to the contents of PDF. The PDF here is Churchill Motor Insurance Policy Booklet. In this project, I utilized HuggingFace's instructor-xl model for embeddings and Google's flan-t5-large model for conversational tasks, ensuring robust and accurate responses

Dataset Construction:-

To ensure that queries are diverse and not concentrated on one topic, I manually curated 32 query response pairs.The queries cover a wide range of topics, ensuring that different sections and types of information are included. Also to ensure that queries are not only of one type, I categorized queries into type of questions like “What”, “Why”, “How” and also whether those are direct queries or indirect queries. Here direct queries mean queries that are present in the PDF directly and whose answer can be accessed directly like FAQs and indirect queries mean question which are not present in the PDF , i.e to find the response the chatbot would have to traverse entire document and summarize it as its answer is not present in the PDF directly.

The dataset includes columns for the query, RAG chatbot response, category, type, human response, and BERT similarity score. Below is a sample of the dataset:



Query	RAG Chatbot Response	Category	Type	Human Response	Score
What is the maximum value you will pay if my car is damaged?	Where damage to your car is covered under your policy, we'll pay the cost of repairing or replacing your car up to its UK market value. This is the current value of your car at the time of the claim.	What	Direct	We'll pay the cost of repairing or replacing your car up to its UK market value at the time of the claim.	4.062110761
Who is covered to drive other cars?	Your certificate of motor insurance will show who has cover to drive other cars. We'll only know about the accident as quickly as possible.	Who	Direct	Your certificate of motor insurance will show who has cover to drive other cars.	3.639629802

Here Score Represents BERT similarity score.
This BERT similarity score is the basis of evaluation for chatbot. (explained in detail in Evaluation section below)

TechStack:-

- Python: Core programming language for development.
- Streamlit: UI development tool for creating interactive web applications.
- PyPDF2, python-docx, pptx: Libraries for parsing and extracting text from document formats.
- PyTesseract: Optical character recognition (OCR) tool for extracting text from images.
- Transformers (Hugging Face): Library for natural language processing (NLP) tasks, including question-answering and image captioning.

- PyTorch: Deep learning framework for tasks such as image captioning using pre-trained models.
- NLTK: Toolkit for natural language processing tasks like tokenization and part-of-speech tagging.
- Hugging Face's google-flan-t5-large: Language model for generating responses to user queries and conducting conversations.
- Hugging Face's: instructor-xl model for embeddings
- Langchain: Library for advanced text processing tasks such as text splitting, embeddings, and conversational chains.

Implementation Summary:-

- The ChatwithPDF project is centered around using advanced language models (LLMs) to create an effective assistant that could summarize your document. The application uses Streamlit for the web interface and LangChain to manage interactions with the LLMs.
- Key functionality is the use of HuggingFace's instructor-xl model for generating text embeddings and Google's flan-t5-large model for handling conversational tasks. These models ensure robust and accurate responses, making the assistant capable of understanding and processing complex pdf documents.
- The workflow involves extracting text from document, splitting the text into manageable chunks using CharacterTextSplitter, and converting these chunks into embeddings. The embeddings are stored in a FAISS vector store for efficient retrieval.
- Here I have also tried for other file formats like PPT, Latex, Word File etc.
- The conversational chain, which uses flan-t5-large, includes a memory buffer to keep track of the chat history, allowing the assistant to provide relevant responses. Users can interact with the assistant by uploading documents and asking questions in natural language, receiving accurate answers based on the content of the documents.
- This integration of LLMs provides a powerful tool for document analysis.
- It takes almost 7-8 mins for getting chunks of this entire policy booklet using Google's flan-t5-large.

Issues / Challenges Faced:-

- Once challenge which I faced was dataset construction of diverse query-response pairs and finding the human response manually so as to compare it with the chatbot response
- Here the main challenge I faced was running the LLM on local machine. Initially, I tried with google-flan-t5-xl which had approximately 11 billion parameters. As a result it was computationally very expensive and time consuming for pdf with more number of pages to run it on a local machine.
- Also using OpenAI led me to challenge of API key.
- I even tried certain other models of Hugging face like Mistral-7B, EleutherAI/gpt-j-6B etc.
  The issue with Mistral-7B was same i.e No of parameters as it had 7 billion parameters and issue with EleutherAI/gpt-j-6B was low performance index.
- Then I switched to google-flan-t5-large which had 728 Million parameters and also better performance index.

Performance Evaluation:-

- The performance of chatbot is evaluated by comparing its response to human response using a pre-trained BERT model from Sentence Transformers.
- A pre-trained BERT model (paraphrase-MiniLM-L6-v2) from Sentence Transformers is used to generate embeddings for both chatbot and human responses.
- Cosine similarity is computed between the corresponding embeddings of chatbot and human responses to measure their similarity.
- Similarity scores are normalized to a scale of 1 to 5 for easier interpretation.
- The number of response pairs with a similarity score above 3.0 is counted.
- The accuracy is calculated by dividing the count of highly similar pairs by the total number of pairs.
- Out of 32 total 25 pairs have similarity scores above 3 and hence we have accuracy of about approximately 80%.

Thus above is the entire documentation for RAG-Based Chatbot.