



















































<https://drive.google.com/drive/folders/1twxYrqAxzDcimg1iFKmsfByA3A9B-6IT?usp=drive_link>

**ProjectScope:**

The objective of this project is to build a **Retriever-Augmented Generation (RAG)** system that can answer medical questions based on information extracted from medical books in PDF format. The system will integrate a retrieval-based model to find the most relevant pieces of information from a large corpus of medical data and a generative language model to synthesize responses based on these retrieved documents. This system will be designed to help medical students, professionals, and researchers quickly access accurate information from a vast range of medical books and references.

The input to the system will consist of medical books in PDF format, which may cover a variety of topics such as anatomy, pharmacology, diseases, treatment methods, and diagnostic procedures. These PDFs will first be processed into text, which will then be split into smaller chunks to facilitate easier retrieval and better performance when searching for relevant information. These chunks will be stored in a vector database, where they can be efficiently searched using embeddings to retrieve the most relevant information based on a user query. The system will return answers that are both accurate and detailed, relying on the generative capabilities of a fine-tuned language model. The goal is to create a robust system capable of answering a wide array of medical queries with high precision and reliability, using content directly extracted from the medical books.

**Algorithms**

The approach to this project involves several key algorithms and techniques that work together to enable efficient retrieval and generation of medical information. The first stage of the project involves text extraction from the medical books. PDF files will be processed using libraries such as PyPDF2, pdfplumber, or PyMuPDF, which allow for the extraction of raw text. Once the text is extracted, it will be split into smaller chunks using algorithms like the **RecursiveCharacterTextSplitter**, which breaks down large documents into manageable sections. This process is necessary to ensure that the system can search efficiently and retrieve relevant parts of the documents rather than processing the entire text at once.

Next, the text chunks will be converted into vector representations using **embedding models** such as sentence-transformers or specialized models like bioBERT or PubMedBERT for medical content. These embeddings capture the semantic meaning of the text and are essential for performing similarity-based retrieval. The vector representations of the text chunks will then be stored in a **vector database** such as **FAISS** or **Pinecone**. These databases are optimized for high-speed similarity searches, allowing the system to quickly retrieve the most relevant chunks of text based on a user's query. The retrieval process is based on **dense retrieval**, where the similarity between the query and stored vectors is calculated using methods like cosine similarity.

Once the relevant chunks of text are retrieved, the system will pass them to a **generative model**, such as **Llama-2** or **GPT-3**, which has been trained or fine-tuned on medical data. This generative model will synthesize a coherent response based on the retrieved documents. The generative model will be used to formulate answers that are medically accurate and contextually appropriate, leveraging the retrieved information to ensure that responses are based on authoritative sources. Fine-tuning the generative model on a specialized medical dataset can improve its ability to generate accurate answers for medical queries.

The entire workflow is built into a **RetrievalQA** system, which integrates the retrieval and generation steps seamlessly. The retriever fetches relevant text chunks, and the generative model produces an answer based on the content of those chunks. This system can be enhanced with additional features like **prompt engineering**, where specific prompts are crafted to guide the LLM in generating more precise responses, or with techniques like **Maximal Marginal Relevance (MMR)**, which balances the relevance and diversity of retrieved results.

Finally, the system will provide an interactive interface, potentially built with **Gradio**, where users can input their queries and receive instant answers. This interface will allow users to interact with the system in a conversational manner, making it accessible and user-friendly for anyone seeking information from the medical books.