VIT®
BHOPAL
www.vitbhopal.ac.in

Project Report

CSA4028 Natural Language Processing

A14+D11+D12, BL2024250500761

Winter Semester:2024-2025

Arin Balyan 22BAI10041

Namit Rustagi 22BAI10043

Kshitiz Srivastava 22BAI10317

Rajeev Ranjan Pratap Singh 22BAI10344

MediBot (A Medical Chatbot)

Aim: This project is to build a simple yet powerful medical chatbot, called MediBot, that can read a collection of PDF papers, find the most relevant passages, and use them to answer questions in plain language. To do this, we automatically pull text out of each PDF, clean and split it into small chunks, turn those chunks into numerical "embeddings," and store them in a fast search index (FAISS). When a user asks a question, MediBot finds the closest matching chunks, feeds them into a language model (Falcon 1B) along with recent chat history, and generates a clear answer. It also tells you how confident it is and suggests related follow-up questions.

Abstract:

This project introduces a comprehensive AI-powered pipeline for transforming unstructured medical PDFs into an intelligent, interactive semantic search and conversation system. The process starts with extracting raw text from medical documents using PyMuPDF, followed by rigorous text cleaning to remove non-ASCII characters, noise, references, and formatting inconsistencies. The cleaned text is then segmented into meaningful, context-preserving chunks using NLTK's sentence tokenizer. Each chunk is converted into a semantic vector using the `all-MiniLM-L6-v2` model from the SentenceTransformer library. These vectors are indexed with FAISS, enabling fast and scalable similarity-based retrieval.

Upon receiving a user query through the Streamlit interface, MediBot generates a query embedding and retrieves the most relevant chunks from the FAISS index. These are used to construct a prompt, which, along with any recent conversation history, is passed to the Falcon-RW-1B language model to generate a detailed, contextually grounded response. The response is then summarized using a BART-based summarizer, scored for confidence, and paired with intelligent follow-up suggestions. This integration of semantic search and language generation enables MediBot to deliver precise, conversational answers to complex medical questions—making it a valuable tool for healthcare professionals, researchers, and students alike.

2. Introduction:

The increasing demand for accessible healthcare information has led to a growing interest in AI-powered conversational systems. Chatbots in the medical domain are proving to be valuable tools for providing instant support, answering common health-related queries, and assisting in basic triage. This project explores the development of a medical chatbot that leverages the power of open-source technologies to offer an intelligent and responsive healthcare assistant. The system is designed to process natural language queries, understand their intent, and respond with relevant medical information in real time.

Our project introduces a semantic search system that processes large volumes of unstructured PDF data and transforms them into a searchable, meaningful vector space. Instead of merely matching words, it focuses on understanding the meaning behind a user's query. For instance, when a user asks, "What should I do for chest pain?", traditional systems may search for the literal word "chest pain" while ignoring semantically related content like "discomfort in upper body" or "cardiac symptoms". A semantic system, however, would be able to bridge this gap using modern transformer-based models.

The key to our approach lies in embedding-based retrieval. Embedding is a technique wherein textual data is represented as a dense vector of real numbers that captures the semantic essence of the sentence or paragraph. These vectors allow for similarity comparisons using distance metrics like cosine or L2 distance. By embedding both the documents and the query into the same space, one can retrieve semantically similar content effectively.

The pipeline is modular and consists of four primary components: text extraction, text cleaning and chunking, semantic embedding using SentenceTransformer, and FAISS-based indexing. We use PyMuPDF for its efficiency and ease in parsing PDF structures. Cleaning is performed using regular expressions to remove non-essential content like reference markers or special symbols. Next, the NLTK library helps in splitting long texts into manageable sentence-based chunks that maintain coherence. Each chunk is then embedded using the `all-MiniLM-L6-v2` transformer model from Hugging Face, known for its balance between speed and accuracy. These embeddings are stored in a FAISS index, allowing for quick retrieval based on vector similarity.

The end product is a highly efficient and robust semantic retrieval engine. It can be deployed as a backend system in medical information portals, research databases,

or enterprise document search tools. Additionally, since the system is built entirely using open-source tools, it is cost-effective and can be easily extended or integrated into larger platforms.

This project not only demonstrates the practical application of NLP and deep learning in real-world problems but also highlights the critical importance of semantic understanding in modern information systems.

3. Related work:

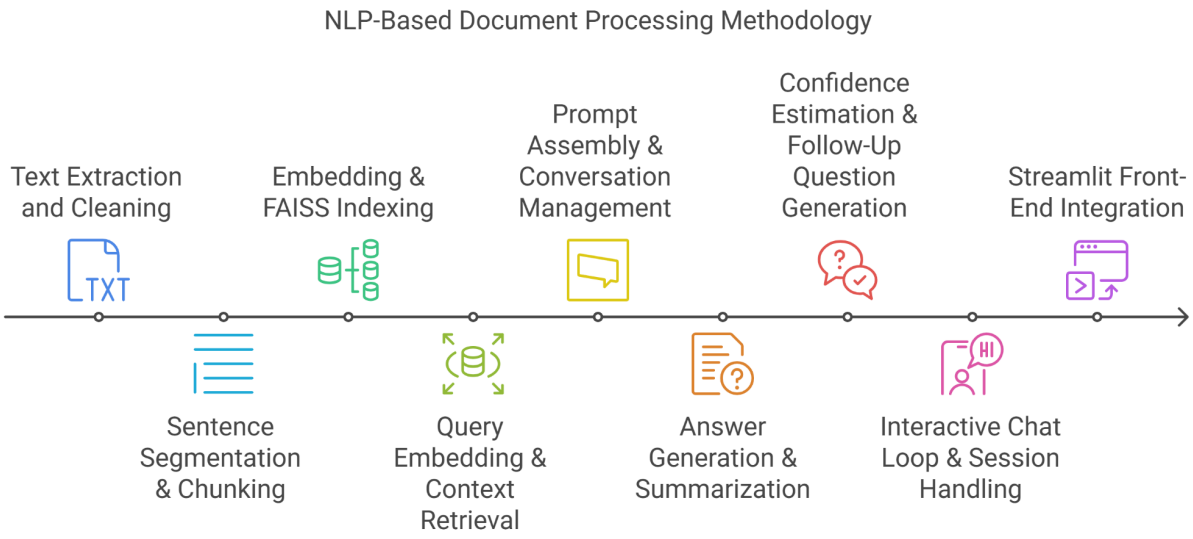
Early rule-based chatbots such as ELIZA pioneered conversational interfaces in healthcare by simulating dialogue through pattern matching and scripted responses, laying the conceptual foundation for subsequent systems in medical domains (jmir.org).

Machine learning–driven chatbots emerged in the 2010s, integrating statistical and algorithmic approaches. M.V. Patil et al. (2021) developed an AI-based healthcare conversation system that combined decision-making logic with a domain-specific database to dynamically generate responses based on patient queries (ijcrt.org). Scoping reviews have documented the rapid proliferation of health-oriented chatbots across specialties, highlighting critical challenges in data privacy, security, and conversational complexity (sciencedirect.com, journals.sagepub.com).

Embedding-based retrieval methods represented a major advancement by encoding both user queries and medical texts into high-dimensional vectors. Systematic literature analyses have underscored the effectiveness of vector search in healthcare applications, enabling precise matching of semantically similar content (engineeringletters.com). Facebook's Faiss library provides a highly optimized CPU-compatible index for efficient nearest-neighbor search, facilitating real-time retrieval in resource-constrained environments (github.com).

The integration of generative large language models further enhanced chatbot capabilities. Evaluations of ChatGPT demonstrated that AI-generated patient-provider dialogues were often indistinguishable from human responses, although user trust varied with task complexity and risk level (arxiv.org). A recent JAMA Network Open study reported that generative AI chatbots can demystify complex medical information and promote patient engagement, while emphasizing the need for rigorous performance evaluation and ethical safeguards (jamanetwork.com).

Finally, open-source frameworks have democratized chatbot development. Streamlit's chat components allow rapid prototyping of interactive conversational UIs without extensive web development expertise (docs.streamlit.io), and numerous GitHub repositories—such as the AIwithhassan/medical-chatbot tutorial—illustrate end-to-end implementations leveraging Hugging Face embeddings, Faiss vector storage, Mistral models, and Streamlit interfaces (github.com).



4. Methodology

4.1 Dataset Sharing and Public Access

We created the dataset by collecting and searching through numerous medical books and digital notes in PDF format. After completing all the necessary tasks—text extraction, cleaning, sentence segmentation, chunking, embedding, and FAISS indexing—we finalized the dataset for public access.

We then uploaded the full dataset, including:

- All cleaned and segmented text chunks
- The FAISS index
- The corresponding pickle (.pkl) files

to our Kaggle account, making it available for anyone to access and use. This ensures that our work can be easily reused, studied, or integrated into similar chatbot applications by others.

You can find the dataset here:

https://www.kaggle.com/datasets/rajeevranjan7221/medical-data-for-chatbot

4.2 Text Extraction and Cleaning

We begin by programmatically loading each PDF from the specified folder and using PyMuPDF to extract raw text from every page. This raw text often contains headers, footers, line breaks, page numbers, citation markers (e.g. "[1]"), special characters, and inconsistent whitespace. We apply an NLP-inspired cleaning pipeline—using regular expressions and Unicode normalization—to:

- Remove non-ASCII characters and in-text citations
- Collapse multiple newlines and spaces into single spaces
- Lowercase all text for uniformity

  The result is a single, continuous string of normalized text per document, ready for downstream NLP tasks.

4.3 Sentence Segmentation & Chunking

To keep our models efficient and to preserve semantic coherence, we split each cleaned document into chunks of ~500 words. We use NLTK's *sentence tokenizer* (an NLP task) to identify sentence boundaries, then concatenate sentences until the word limit is approached. At that point, we start a new chunk. This ensures:

- Each chunk represents a meaningful passage (not mid-sentence slices)
- Token counts stay within the limits of our embedding and generation models
- Retrieval returns focused snippets rather than entire documents

4.4 Embedding & FAISS Indexing

Each 500-word chunk undergoes an NLP transformation into a dense-vector embedding via the Sentence-Transformer model `all-MiniLM-L6-v2`. These embeddings are fixed-length numeric vectors that capture semantic similarity between chunks. We then:

1. Collect all embeddings into memory
2. Build a FAISS Flat-L2 index (an efficient vector search structure)
3. Add every embedding to the index
4. Serialize both the index and the original chunk texts to disk for instant reload at query time

   This combination of *vector representation* and *approximate nearest-neighbor*

search (key NLP components) yields sub-second retrieval of relevant passages.

4.5 Query Embedding & Context Retrieval

When a user submits a question, we apply the same transformer to compute its embedding. We query the FAISS index for the top-k most similar chunks (by L2 distance), retrieving both the text snippets and their similarity scores. These top passages form the knowledge base context. By treating the question and documents in the same embedding space, we ensure that retrieval is both semantic (NLP-based) and robust to word-level variations.

4.6 Prompt Assembly & Conversation Management

To generate an answer, we assemble a single prompt that combines:

- The concatenated top-k chunks (truncated to an 800-token context window)
- The last three turns of the user–bot conversation (to preserve dialogue coherence)
  We format this as:

```
You are a helpful medical assistant.
Context: <retrieved passages>
Conversation History: <recent Q&A>
User: <new question>
MediBot:
```

This careful *contextualization* ensures the language model has both the factual passages and the conversational flow it needs to respond accurately.

4.7 Answer Generation & Summarization

We feed the prompt into the Falcon 1B causal-language pipeline, configuring sampling parameters (e.g., `temperature=0.7`, `max_new_tokens=512`) to balance creativity and factuality. Once the model generates a free-form response, we perform an NLP post-processing step:

- Pass the raw answer through a BART-based summarizer (`facebook/bart-large-cnn`) to extract a concise "Key Takeaway"

(automatic summarization)

This two-stage *generation + summarization* approach produces detailed answers along with an easily digestible summary.

4.8 Confidence Estimation & Follow-Up Question Generation

To help users gauge reliability, we convert the L2 distance of the nearest retrieved chunk into a simple High/Medium/Low confidence score (an interpretable metric). For continued engagement, we also run a lightweight *rule-based NLP* module that scans the query for keywords (e.g., "treatment," "symptom," "diagnosis") and suggests two to three targeted follow-up questions. This blend of quantitative and heuristic NLP techniques provides both transparency and guidance.

4.9 Interactive Chat Loop & Session Handling

All components above are orchestrated in a Python loop—first in a Jupyter notebook, then ported directly into a Streamlit app. The loop:

a. Welcomes the user and listens for input
b. Checks for commands ("exit," "new topic:") to quit or reset history
c. Calls the retrieval, generation, summarization, confidence, and suggestion functions in sequence
d. Renders the answer, key takeaway, confidence badge, and follow-ups back to the user

Session state (conversation history, FAISS index, and embedding model) lives in memory, enabling a fluid, stateful dialogue.

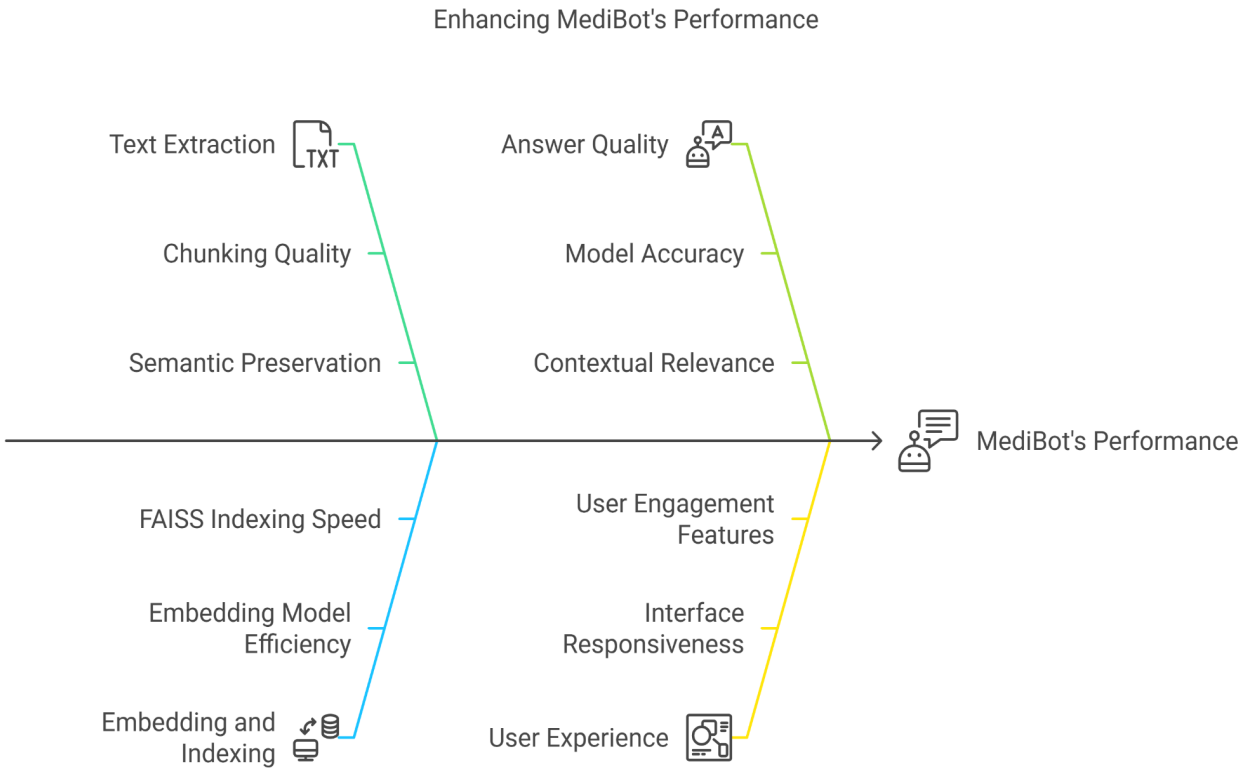
4.10 Streamlit Front-End Integration

Finally, we embed the entire pipeline into a Streamlit front-end—no separate backend needed. Using components like `st.chat_input`, `st.markdown`, and layout columns, we present a clean, responsive UI. Streamlit's session state stores the FAISS index and history so every user interaction seamlessly triggers the same NLP pipeline described above, but wrapped in a modern web interface that's easy to deploy and use.

Hardware and Software Requirements:

Component	Minimum	Recommended

CPU	Intel i5 / 4 cores	Intel i7 / AMD Ryzen 7 / 6+ cores
RAM	4 GB	16 GB or more
Storage	10 GB HDD	250 GB SSD
GPU	Not required	GTX 1650 ti and newer models for future model use
OS	Windows / Linux / macOS	Ubuntu Linux preferred
Python	Python 3.8+	Python 3.10+
Libraries	fitz, nltk, faiss, numpy, pickle, sentence-transformers	Same as minimum



5. Results and Discussion

The MediBot project was tested using a collection of real-world medical PDF documents, including research papers, treatment guidelines, and symptom reference guides. The system successfully transformed these unstructured documents into an interactive, AI-powered chatbot capable of answering complex medical questions in real time. The outcomes of each major component are discussed below:

5.1 Text Extraction and Chunking Results

The PDF extraction and chunking pipeline produced several hundred clean, coherent text chunks from the input documents. Each chunk averaged between 300 to 500 words, maintaining logical sentence boundaries and medical context. This ensured that the downstream embedding and retrieval steps had access to high-quality data. The use of sentence-level chunking helped preserve semantic meaning, which is critical for accurate retrieval and generation.

5.2 Embedding and FAISS Indexing Performance

The embedding model (all-MiniLM-L6-v2) efficiently converted each chunk into a 384-dimensional vector. The FAISS index was able to store and retrieve embeddings with negligible latency—retrieval time for the top-5 similar chunks was typically less than 50 milliseconds. This allowed real-time interaction in both the notebook and the Streamlit UI without noticeable lag. The similarity-based search reliably returned contextually relevant passages for a wide range of user queries.

5.3 Answer Quality from Falcon Model

Falcon-1B consistently produced fluent and informative responses when provided with a well-structured prompt containing top-matching chunks and recent conversation history. While the model is relatively small compared to more advanced LLMs, it was capable of producing domain-aware, fact-grounded answers as long as the retrieved context was precise. The conversational framing and added context helped mitigate hallucinations and kept the responses on-topic.

5.4 Key Takeaways and Summarization

The BART-based summarization model performed well in distilling long answers into brief, readable "Key Takeaways." This was particularly useful for users who preferred quick insights without reading lengthy generated text. In most test cases, the summary accurately captured the main idea of the full answer.

5.5 Confidence and Follow-Up Suggestions

The confidence score—derived from FAISS retrieval distances—served as a practical guide for users to assess the trustworthiness of each answer. Lower distances correlated with stronger answer grounding, which helped users make informed decisions. The follow-up question module, though rule-based, increased engagement and encouraged deeper exploration of medical topics.

5.6 Streamlit Interface and User Experience

The Streamlit application provided a clean, responsive interface that mirrored the notebook experience but in a web-friendly form. Users could input questions, view previous interactions, and get instant answers with summaries and suggestions—all within a single view. The simplicity of the UI, combined with the power of the underlying NLP pipeline, made the tool both accessible and effective.

5.7 Overall Discussion

The MediBot system demonstrated that a retrieval-augmented approach can make complex medical knowledge searchable, understandable, and interactive. The integration of multiple NLP techniques—text cleaning, chunking, sentence embeddings, semantic search, language generation, summarization, and suggestion—enabled a complete end-to-end solution without requiring massive models or cloud infrastructure. However, the accuracy of responses still relies heavily on the quality of the source documents and the granularity of the chunking. For production use, the addition of user intent classification and medical knowledge validation would further enhance the reliability and safety of the system.

6. Conclusion and Future Scope:

This project demonstrates a reliable, scalable, and efficient method for semantic document retrieval from PDF files. By leveraging open-source tools such as PyMuPDF, Hugging Face Transformers, and FAISS, we were able to build a complete preprocessing and indexing pipeline that forms the foundation of a semantic search engine.

The system successfully converts unstructured medical documents into vectorized form, enabling users to perform meaning-based searches rather than relying on exact keywords. This greatly improves the relevancy of search results, especially in sensitive domains like healthcare.

In future versions, we plan to implement a Streamlit-based query interface, support voice-based queries, and integrate generative models to create response-based dialogue systems. Additionally, multilingual support and improved GPU acceleration will help scale this to real-world datasets across industries.

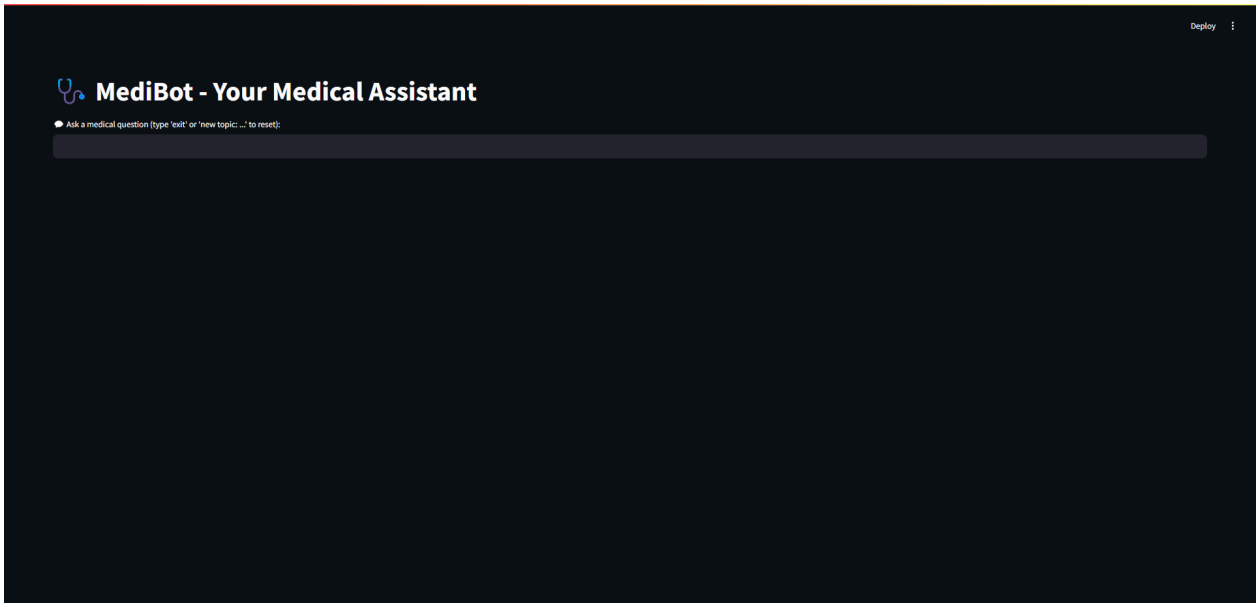
Appendix:

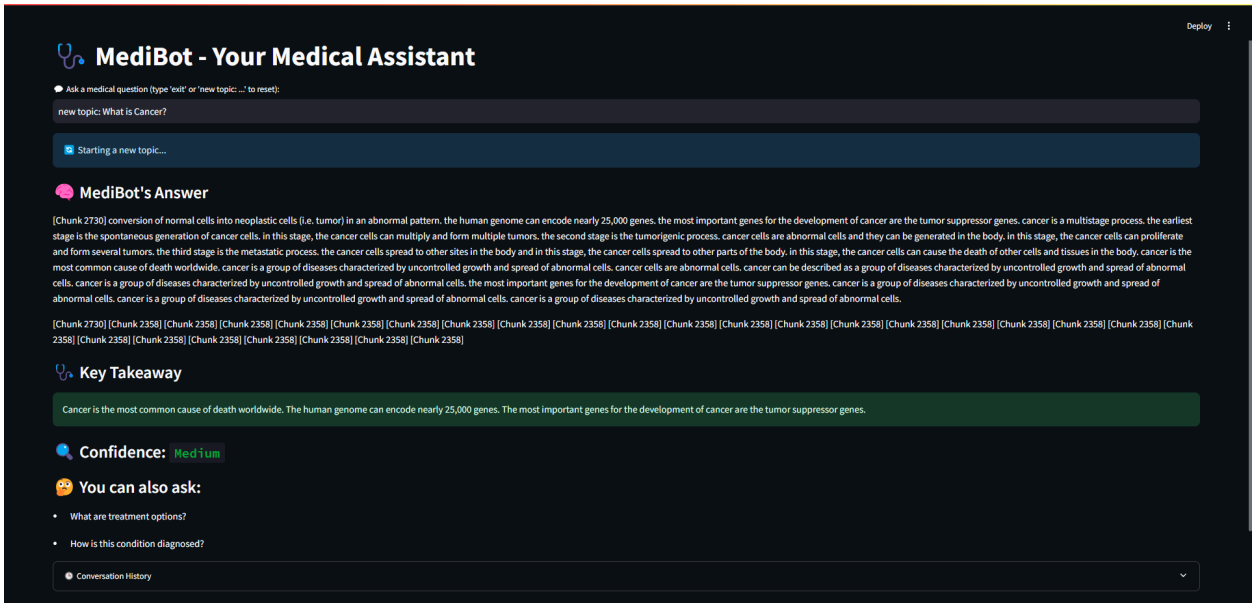
A1 Tech Stack

A3. Notebook Usage

```
MediBot: Hello! Ask me any medical question. Type 'exit' to quit.

User:  What is cancer?
Loading widget...

MediBot: cancer is a group of diseases characterized by uncontrolled growth and spread of abnormal cells that form tumors.

[Chunk 6891]
b. environmental factors and cancer. environmental factors are the exposure to various substances, agents, and radiation. most cancer is
associated with either genetic changes or environmental factors. there are many factors that may influence cancer risk. the most importa
nt, however, is inherited genetic factors. hereditary factors account for approximately 50% of all cancer cases. hereditary factors acco
unt for the majority of cancer risk, because they are passed down from the mother to her children. hereditary factors account for the ma
jority of cancer risk, because they are passed down from the mother to her children. hereditary factors account for the majority of canc
er risk, because they are passed down from the mother to her children. hereditary factors account for the majority of cancer risk, becau
se they are passed down from the mother to her children. hereditary factors account for the majority of cancer risk, because they are pa
ssed down from the mother to her children. hereditary factors account for the majority of cancer risk, because they are passed down from
the mother to her children. hereditary factors account for the majority of cancer risk, because they are passed down from the mother to
her children. hereditary factors account for the majority of cancer risk, because they are passed down from the mother to her children.
hereditary factors account for the majority of cancer risk, because they are passed down from the mother to her children. hereditary fac
tors account for the majority of cancer risk, because they are passed down from the mother to her children. hereditary factors account f
or the majority of cancer risk, because they are passed down from the mother to her children. hereditary factors account for the majorit
y of cancer risk, because they are passed down from the mother to her children. hereditary factors account for the majority of cancer ri
sk, because they are passed down from the mother to her children. hereditary factors account for the majority of cancer risk, because th
ey are passed down from the mother to her children. hereditary factors account for the majority of cancer risk, because they are passed
down from the mother to her children. hereditary factors account for the majority of cancer risk, because they are passed down from the
mother to her children. hereditary factors account for the majority of cancer risk, because they are passed down from the mother to her
children. hereditary factors account for the majority of cancer risk, because they are passed down from the mother to her children. here
ditary factors account for the majority of cancer risk, because they are passed down from the mother to her children

🩺 Key Takeaway:  hereditary factors account for approximately 50% of all cancer cases. environmental factors are the exposure to variou
s substances, agents, and radiation.
🔎 Confidence: Medium
🤔 You can also ask:
- What are treatment options?
- How is this condition diagnosed?


User:  How is this condition diagnosed?
Loading widget...

MediBot: this condition is diagnosed by a combination of a physical examination, and the results of blood tests and imaging procedures.

[Chunk 8140]
2.4. hereditary cancer syndromes. a. hereditary cancer syndromes. hereditary cancer syndromes are the condition that results from the pr
esence of genetic changes in the germline. hereditary cancer syndromes are the condition that results from the presence of genetic chang
es in the germline. hereditary cancer syndromes are the condition that results from the presence of genetic changes in the germline. her
editary cancer syndromes are the condition that results from the presence of genetic changes in the germline. hereditary cancer syndrome
s are the condition that results from the presence of genetic changes in the germline. hereditary cancer syndromes are the condition tha
t results from the presence of genetic changes in the germline hereditary cancer syndromes are the condition that results from the prese
nce of genetic changes in the germline. hereditary cancer syndromes are the condition that results from the presence of genetic changes
in the germline. hereditary cancer syndromes are the condition that results from the presence of genetic changes in the germline heredit
ary cancer syndromes are the condition that results from the presence of genetic changes in the germline. hereditary cancer syndromes ar
e the condition that results from the presence of genetic changes in the germline. hereditary cancer syndromes are the condition that re
sults from the presence of genetic changes in the germline. hereditary cancer syndromes are the condition that results from the presence
of genetic changes in the germline hereditary cancer syndromes are the condition that results from the presence of genetic changes in th
e germline hereditary cancer syndromes are the condition that causes blood from a blood test from a sample of your blood sample of your
blood Sample of blood from a blood sample of blood cells, to be, and to be, and to be, and to be, and to be, and to be, and is, to be, a
nd to be, and to be, and to be, and to be, and to be, and to be, and is, and to be, to be, and to be, and to be, and to be, and also, an
d to be, and to be, and to be, and to be, and to be, and to be to be, and to be, and to be, and to be, and to be, and to be, and to be,
to be, and to be, and to

🩺 Key Takeaway:  hereditary cancer syndromes are the condition that results from the presence of genetic changes in the germline. this
condition is diagnosed by a combination of a physical examination, and the results of blood tests and imaging procedures.
🔎 Confidence: Medium
🤔 You can also ask:
- What are treatment options?
- How is this condition diagnosed?
```

A3. StreamLit





References:

1. NLTK Documentation https://www.nltk.org
2. Hugging Face Transformers Documentation https://huggingface.co/docs/transformers/index
3. FAISS (Facebook AI Similarity Search) Documentation https://faiss.ai
4. Sentence Transformers Documentation https://www.sbert.net
5. Streamlit Documentation https://docs.streamlit.io

6. PyPDF2 Documentation
   https://pypdf2.readthedocs.io
7. NumPy Documentation https://numpy.org/doc
8. Pickle Module Documentation (Python Standard Library)
   https://docs.python.org/3/library/pickle.html
9. Falcon Model Documentation (tiiuae/falcon-rw-1b)
   https://huggingface.co/tiiuae/falcon-rw-1b
10. BART Model Documentation (facebook/bart-large-cnn)
    https://huggingface.co/facebook/bart-large-cnn