Medical AI ChatBot-Focusing on Gynecology

DATS 6312 Natural Language Processing - Final Project- Group 3

Individual Report- Tanmay Ambegaokar

I. Introduction

1.1 Overview

Our final project addresses the challenge of fine-tuning a Large Language Model (LLM) for application within the medical domain, with a specific focus on developing expertise in a specialized subfield. The key components of the project are as follows:

1. Topic Modeling: Utilizing topic modeling techniques, we extracted the top 10 topics from the features of the dataset. Based on frequency analysis, we selected a specific domain of interest—pregnancy—as the focus area for specialization.

2. Data Retrieval: A corpus of 10 comprehensive books on the topic of pregnancy was identified and their digital (PDF) versions were obtained. This, along with Wikipedia data on the topics forms the knowledge base for the Retrieval-Augmented Generation (RAG) system.

3. Fine-Tuning the LLM: Llama 3.2 (1B parameters) was chosen as the base LLM due to its accessibility and compatibility with our computational resources. This model was fine-tuned using the AI Medical Chatbot dataset, ensuring its alignment with medical conversational tasks.

4. Retrieval-Augmented Generation (RAG): The fine-tuned model was integrated with a RAG framework, creating a specialized medical assistant. This system is designed to emulate a virtual doctor with expertise in gynecology, offering accurate and contextually relevant responses within the domain of pregnancy.

By integrating these components, our project aims to demonstrate the potential of domain-specialized LLMs in delivering expert-level assistance

in critical areas of healthcare.
The AI Medical Chatbot data can be found here. The dataset contains Patient questions and Doctor's responses to them along with a brief description of the actual query of the patient.

1.2 Shared Work

Raghav and I initiated the process by conducting topic modeling to identify a specific focus area for model specialization. We further explored and gathered relevant data sources for this topic.

Subsequently, the fine-tuning phase was undertaken by Parv, Raghav, and Sajan, refining the model using the selected dataset.

Development of the Retrieval-Augmented Generation (RAG) system was done by Parv, Raghav, and I, while the user interface was designed and implemented on Streamlit by Parv and me.

II. Individual Work

2.1 Overview of Individual Work

Once we finalized the dataset, the next step involved analyzing it to uncover key trends and patterns. To gain a deeper understanding, we decided to employ topic modeling to identify the types of questions frequently asked to doctor. This analysis revealed a significant number of queries related to pregnancy and gynecology, indicating a strong user interest in this domain.

Based on these findings, we decided to focus our efforts on extracting more information and curating additional data specifically on pregnancy and gynecology. We extracted data from two key sources: Wikipedia and authoritative books on pregnancy and gynecology commonly used by both medical students and patients. This comprehensive dataset provided a rich foundation of knowledge on the topic.

Next, we processed these texts by converting them into embeddings and storing them in a Pinecone database to enable efficient retrieval. To enhance the chatbot's performance, we utilized the fine-tuned LLM, Llama 3.2, which delivered improved results by effectively leveraging the

RAG context for generating accurate and context-aware responses.
Finally, I took charge of designing and implementing the user interface on Streamlit, ensuring a seamless and user-friendly experience for interacting with the chatbot.

2.2 Explanation

1. Topic Modelling

We conducted topic modelling using Latent Semantic Analysis (LSA) to uncover key themes in the dataset.

Process:

- **Text Preprocessing**: We tokenized the text, removed stop words. Since there were a lot of garbage words, I used custom stop words to make sure those words do not come up as topics. Lastly, the text was lemmatized

- **TF-IDF Vectorization**: Text was converted into a numerical representation using TF-IDF, which assigns importance to terms based on their frequency and uniqueness across documents.

- **LSA with SVD**: We applied Singular Value Decomposition (SVD) to reduce the dimensionality of the TF-IDF matrix, identifying 10 topics. Each topic was represented by a cluster of significant terms, with importance measured by singular values.

Insights:
The analysis revealed 10 topics, with a prominent focus on pregnancy and gynecology, which guided our decision to extract more targeted data for our RAG.

2. Data Extraction

I identified 10 books which are great books for doctors and for new expectant mothers as our source. I used LangChain's PDFPlumber Document Loader for efficient data extraction from authoritative books on pregnancy and gynecology.

Process:

- **Document Loading**: Using the PDFPlumber Document Loader from LangChain, we loaded the PDF versions of the books, effectively handling complex document structures like headers and multi-column layouts.

- **Text Extraction:** The loader extracted text page by page, accurately capturing content such as paragraphs and headings while ignoring non-informative elements like page numbers and decorative elements.

The extracted text was stored as raw.txt files for further processing and embedding generation.

3. **RAG Implementation**

For the RAG implementation, we processed the raw.txt files into embeddings using the **WhereIsAI/UAE-Large-V1** model. The text was segmented into a chunk size of 1500 with chunk overlap of 100 and converted into high-dimensional embeddings, which were stored in a **Pinecone vector database** along with metadata.

During queries, relevant embeddings were retrieved from Pinecone and passed as context to the fine-tuned **Llama 3.2** model, enabling accurate and context-aware responses. This streamlined pipeline ensured efficient retrieval and precise chatbot answers.

4. S**treamlit**

I developed a user-friendly interface using **Streamlit** to enable seamless interaction with the chatbot. The app featured a clean layout with a query input box and dynamic response display, integrating with the RAG pipeline for real-time, context-aware answers. Additional features, like a sidebar for topic-specific queries and source context display, enhanced usability and engagement.

2.3 Results

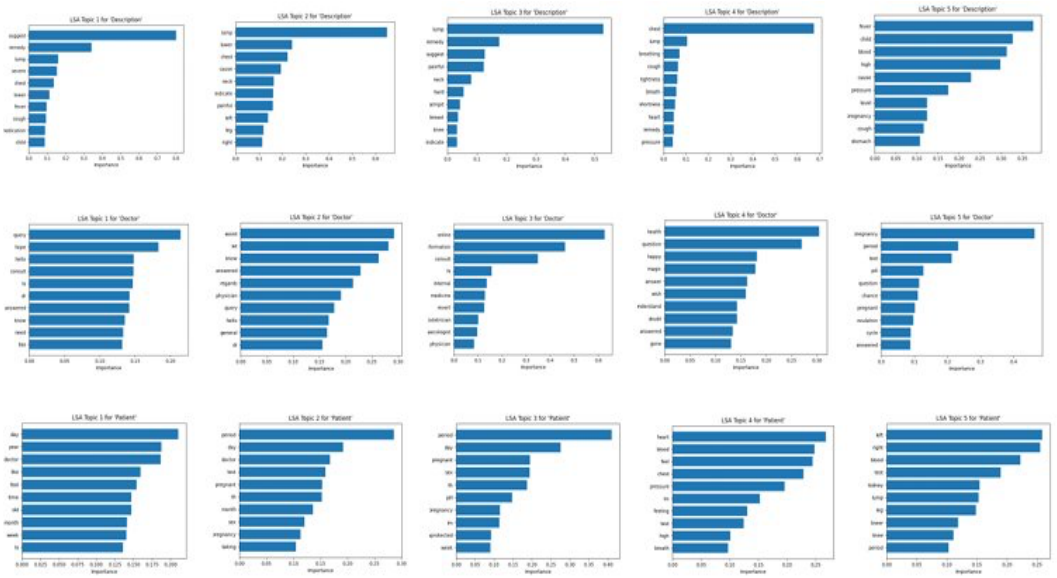
1. **Topic Modelling**

Here’s a look at our original dataset:

Description string · *lengths* 1↔152 99.8%	Patient string · *lengths* 1↔1.78k 99.3%	Doctor string · *lengths* 2↔1.14k 95.1%
Q. Will Nano-Leo give permanent solution for erection problem?	Hello doctor, I am 48 years old. I am experiencing weak erection and…	Hi. For further doubts consult a sexologist online -->
Q. Every time I eat spicy food, I poop blood. Why?	Hi doctor, I am a 26 year old male. I am 5 feet and 9 inches tall and…	Hello. I have gone through your information and test reports…
Q. Will Kalarchikai cure multiple ovarian cysts in PCOD?	Hello doctor, I have multiple small cysts in both ovaries (PCOS). Our…	Hello. I just read your query. See Kalarachi Kai choornam is helpful i…
Q. I masturbate only by rubbing the tip of the penis. Is it a…	Hi doctor, During masturbation I just rub the tip of the penis and…	Hi. For further doubts consult a sexologist online -->
Q. How to calculate the number of days of pregnancy?	Hi doctor, I am 18 weeks 2 days pregnant. My friend told me that I…	Hi. For further information consult an obstetrician and gynaecologist…
Q. As I am having a history of miscarriages, will taking hCG…	Hello doctor, My lmp was on five months ago. I was spotting for thre…	Hello. Unfortunately what your doctor has said is right. Spotting…

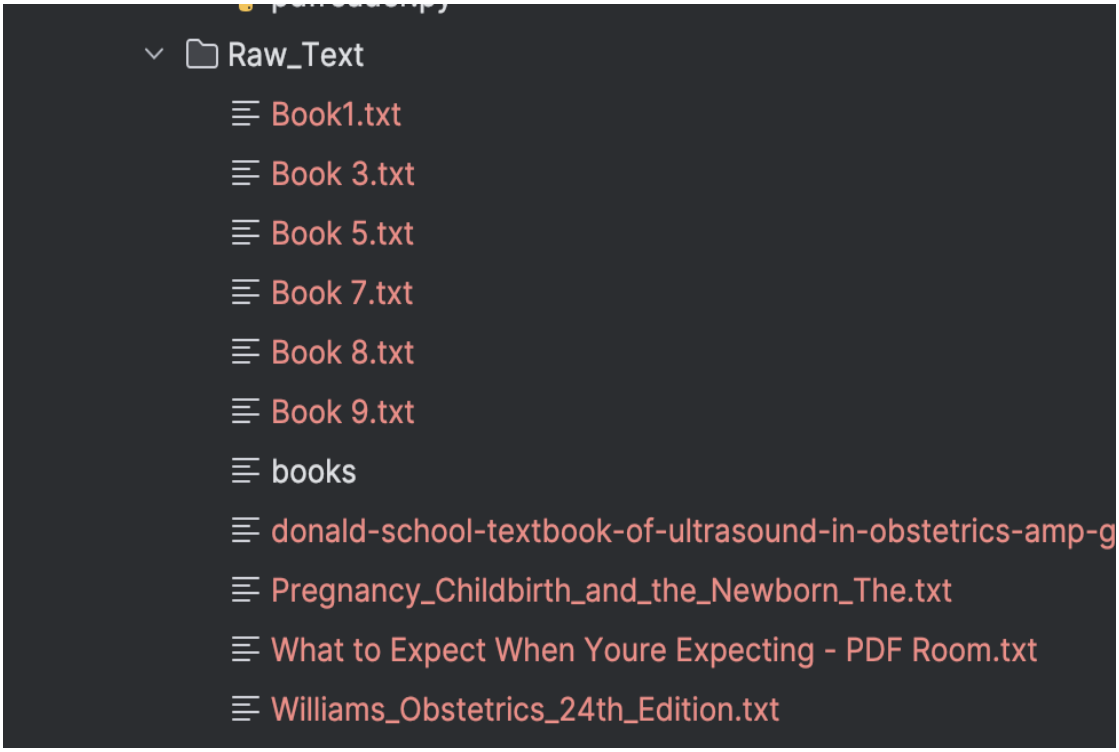
‹ Previous 1 2 3 ... 2,570 Next ›

From this dataset, we did topic modelling and got the most important topics in the dataset with their importance.



2. Data Extraction

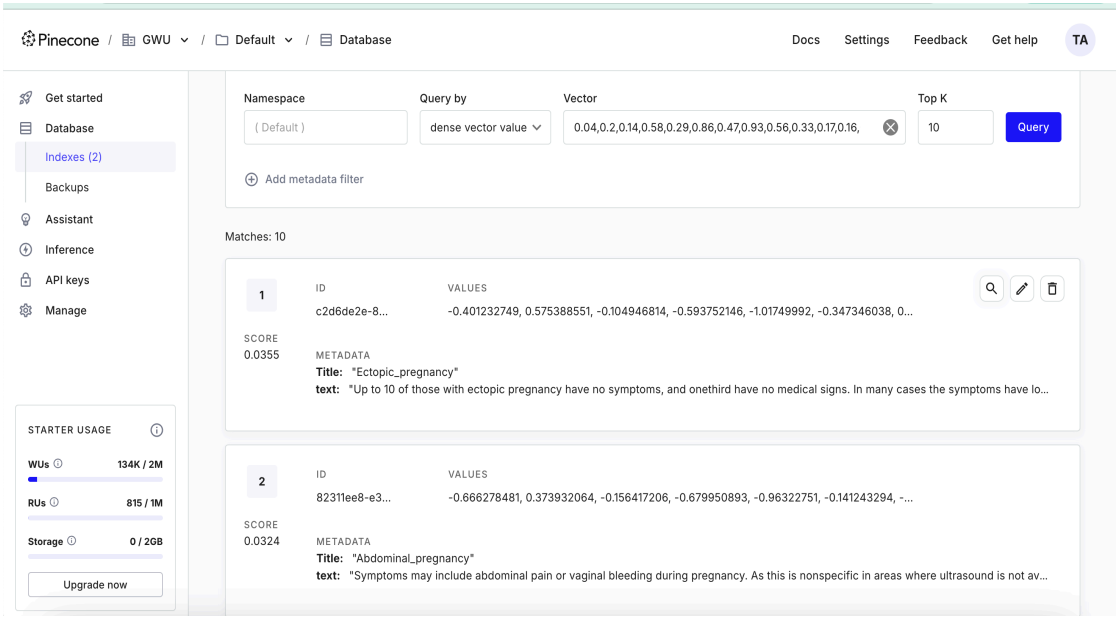
I downloaded 10 textbooks. Then these were processed to raw text files using LangChain's internal PDF document loaders and saved as text files.



3. RAG Implementation

The chunk size was kept at 1500 with an overlap of 100. It took roughly around 3 hours for each text file to be chunked and upserted into the Pinecone database.

Here's a look at the embeddings with meta data and text:



To implement RAG with our fine-tuned model, I did prompt engineering to make sure the model searches for the current context

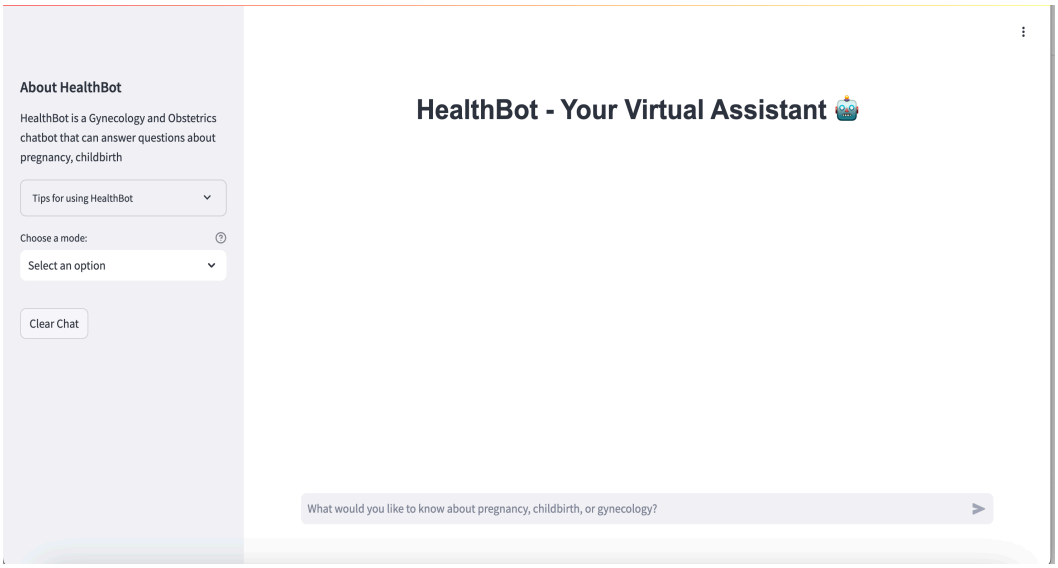
when the user inputs the question.

```
 #Retrieve relevant context using the vector DB
vector_db = VectorDB(
    pinecone_api_key=os.getenv("PINECONE_API_KEY"),
    pinecone_env=os.getenv("PINECONE_ENV"),
    index_name=os.getenv("PINECONE_INDEX_NAME"),
    dimension=int(os.getenv("DIMENSION")),
    metric=os.getenv("METRIC"),
    cloud=os.getenv("PINECONE_CLOUD")
)

# Get the top-k relevant context documents from the vector DB
search_results = vector_db(query, top_k=5)
context = ""
for result in search_results.get('matches', []):
    if 'metadata' in result and 'text' in result['metadata']:
        context += result['metadata']['text'] + "\n"

if not context:
    context = "No relevant context found."


prompt = f"""
Question: {query}
Context: {context}
Your Response as an expert doctor specializing in gynecology, obstetrics, and pregnancy:

"""
```

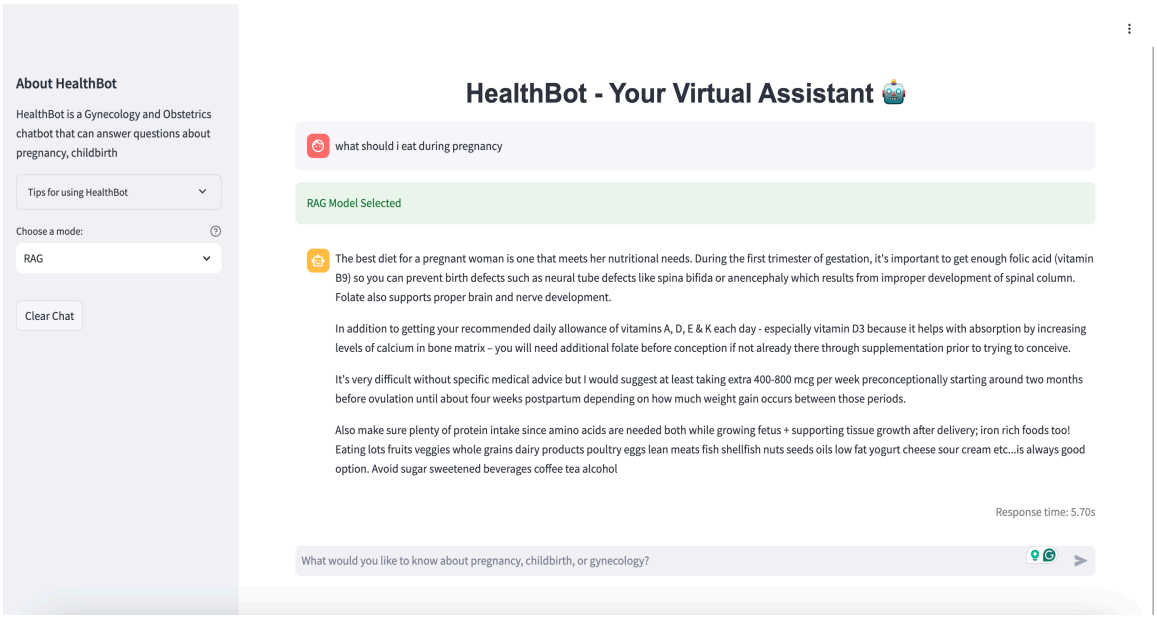
4. Streamlit

Lastly, I made the UI for our app. I handled errors using warnings and success messages for the user's convenience.

Here's a look at the landing page:



Here's a look at one of the results from RAG implemented output:



III. Summary and Conclusion

This project focused on building a Retrieval-Augmented Generation (RAG) chatbot specialized in pregnancy and gynecology. We began by performing topic modelling using LSA with TF-IDF, which revealed a significant focus on questions in this domain. Based on these insights, data was extracted from reliable sources like books and Wikipedia using LangChain's PDFPlumber loader and stored as raw text files. These files were processed into embeddings using the **WhereIsAI/UAE-Large-V1** model and stored in Pinecone for efficient retrieval.

The chatbot leveraged these embeddings as context with the fine-tuned **Llama 3.2** model to generate accurate and relevant answers. The user interface was developed using Streamlit, providing a clean, responsive design for seamless interaction.

To evaluate the chatbot's performance, we employed a human-in-the-loop approach, where we ourselves assessed the accuracy and relevance of the responses. This iterative feedback process ensured continuous improvement in the chatbot's reliability and contextual understanding.

By combining robust data processing, state-of-the-art models, and an intuitive interface, this project successfully created an effective and user-friendly tool for addressing specialized queries in the medical domain.

IV.Code Percentage: 64%

V. References

i. https://huggingface.co/datasets/ruslanmv/ai-medical-chatbot
ii. https://python.langchain.com/docs/how_to/document_loader_pdf/
iii. https://python.langchain.com/docs/integrations/tools/wikipedia/
iv. https://www.pinecone.io/product/
v. https://huggingface.co/WhereIsAI/UAE-Large-V1
vi. https://angle.readthedocs.io/en/latest/index.html
vii. https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05
viii. https://aws.amazon.com/what-is/retrieval-augmented-generation/
ix. https://www.deeplearning.ai/short-courses/langchain-chat-with-your-data/
x. https://www.deeplearning.ai/short-courses/vector-databases-embeddings-applications/
xi. https://github.com/amir-jafari