MEDICAL CHABOT

Revolutionizing Healthcare with AI: A Comprehensive Overview of our Medical Chatbot Project

Prepared by:
Rayan Al Zahabi

CONTENTS

Introduction 3
Problem Definition 3
The Challenge in Healthcare Communication 3
How the Problem Has Been Previously Solved 3
Limitations of Existing Solutions 3
Final Proposed Approach 3
Innovative Chatbot Infused with Advanced Technologies 3
Introduction 4
Process Flow 6
Experimental Setup 7
Results, Discussion, and Outcomes 7
Unveiling the Power of the Medical Chatbot 7
Implications of Results on Healthcare 8
Potential Directions for Future Work and Improvements 8
Continuing the Innovation Journey 8
Conclusion 8
References 9

INTRODUCTION

In recent years, the intersection of artificial intelligence (AI) and healthcare has paved the way for innovative solutions to longstanding challenges. This blog post explores our journey in developing a medical chatbot—a cutting-edge AI project designed to assist users with inquiries about various diseases. In this post, we delve into the problem definition, existing solutions, our novel approach, experimental setup, results, implications, and future directions.

PROBLEM DEFINITION

The Challenge in Healthcare Communication

Effective communication in healthcare is pivotal, yet traditional methods may not always be accessible or user-friendly. Patients often find it challenging to gather information about diseases quickly. Our goal was to create an AI-powered chatbot capable of providing accurate and timely responses to user queries about various medical conditions.

HOW THE PROBLEM HAS BEEN PREVIOUSLY SOLVED

Limitations of Existing Solutions

Traditional healthcare information retrieval systems often lack the speed and accessibility demanded by users. Manual search processes or static databases may not keep up with the dynamic nature of medical knowledge. This inspired us to seek a more efficient and interactive solution that leverages the power of AI.

FINAL PROPOSED APPROACH

Innovative Chatbot Infused with Advanced Technologies

To overcome the limitations of current solutions, we engineered a resilient chatbot. Our strategy intertwines cutting-edge natural language processing (NLP) techniques, Hugging Face embeddings, and a custom-trained language model—LLAMA. At the core of our system is the seamless integration of a document vector store, empowered by FAISS and orchestrated through the LangChain framework. This amalgamation ensures not only the accuracy but also the efficient retrieval of pertinent medical information, marking a significant advancement in chatbot capabilities.

Using LLaMA 2.0, FAISS and LangChain for Question-Answering on our Own Data

Over the past few weeks, we have been playing around with several large language models (LLMs) and exploring their potential with all sorts of methods available on the internet, but now it's time for us to share what we have learned so far!

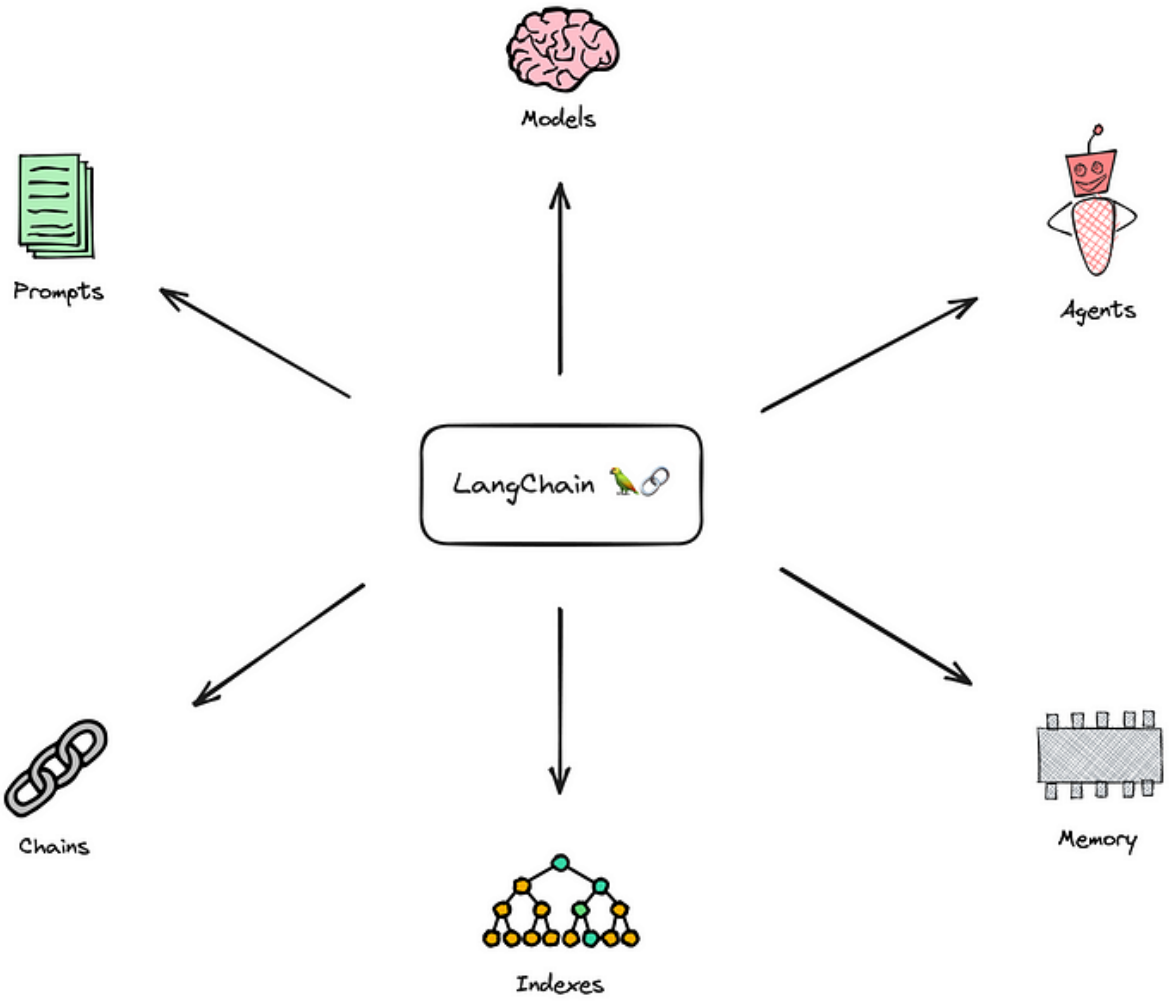
We were super excited to know that Meta released the next generation of its open-source large language model, LLaMA 2 (on 18th July 2023) and the most interesting part of the release was, they made it available free of charge for commercial use to the public. Therefore, we decided to try it out and see how its performs.

We are going share on how we performed Question-Answering (QA) like a chatbot using **Llama-2–7b-chat** model with **LangChain** framework and **FAISS library** over our documents.

Introduction

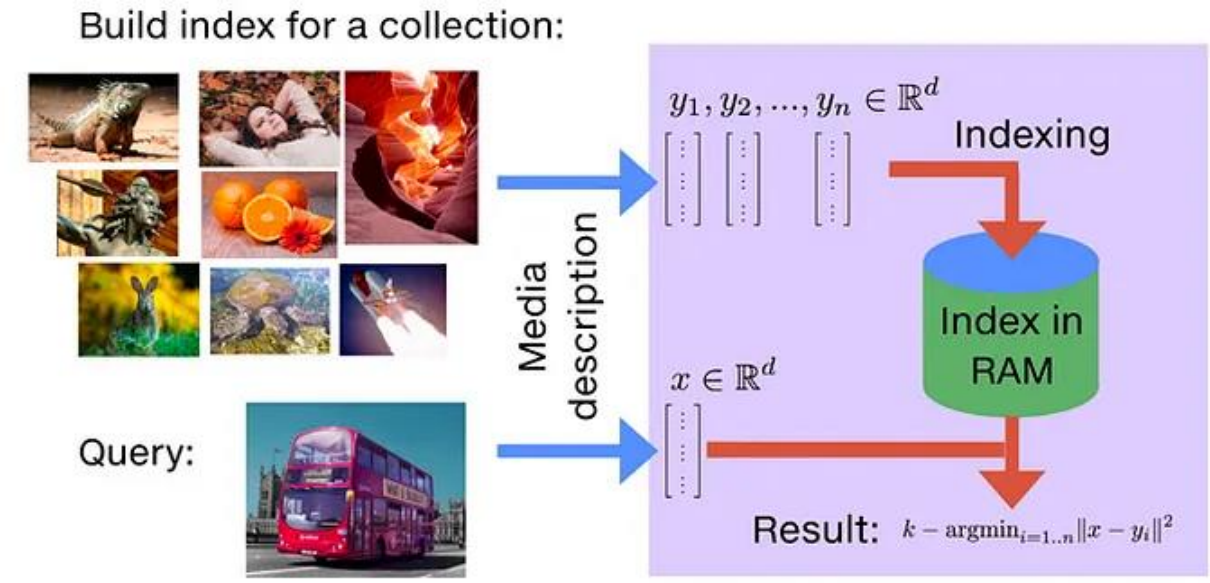
- **LLaMA 2** model is pretrained and fine-tuned with 2 Trillion 🚀 tokens and 7 to 70 Billion parameters which makes it one of the powerful open source models. It comes in three different model sizes (i.e. 7B, 13B and 70B) with significant improvements over the Llama 1 models, including being trained on 40% more tokens, having a much longer context length (4k tokens 🤯), and using grouped-query attention for fast inference of the 70B model 🔥. It outperforms other open source LLMs on many external benchmarks, including reasoning, coding, proficiency, and knowledge tests.
  Note: We used quantized Llama 2 Model due our limited resources (CPU and GPU) on our laptop.
  Link to our model: https://huggingface.co/TheBloke/Llama-2-7B-Chat-GGML/blob/main/llama-2-7b-chat.ggmlv3.q8_0.bin

- **LangChain** is a powerful, open-source framework designed to help you develop applications powered by a language model, particularly a large language model (LLM). The core idea of the library is that we can “chain” together different components to create more advanced use cases around LLMs. LangChain consists of multiple components from several modules.



LangChain Modules

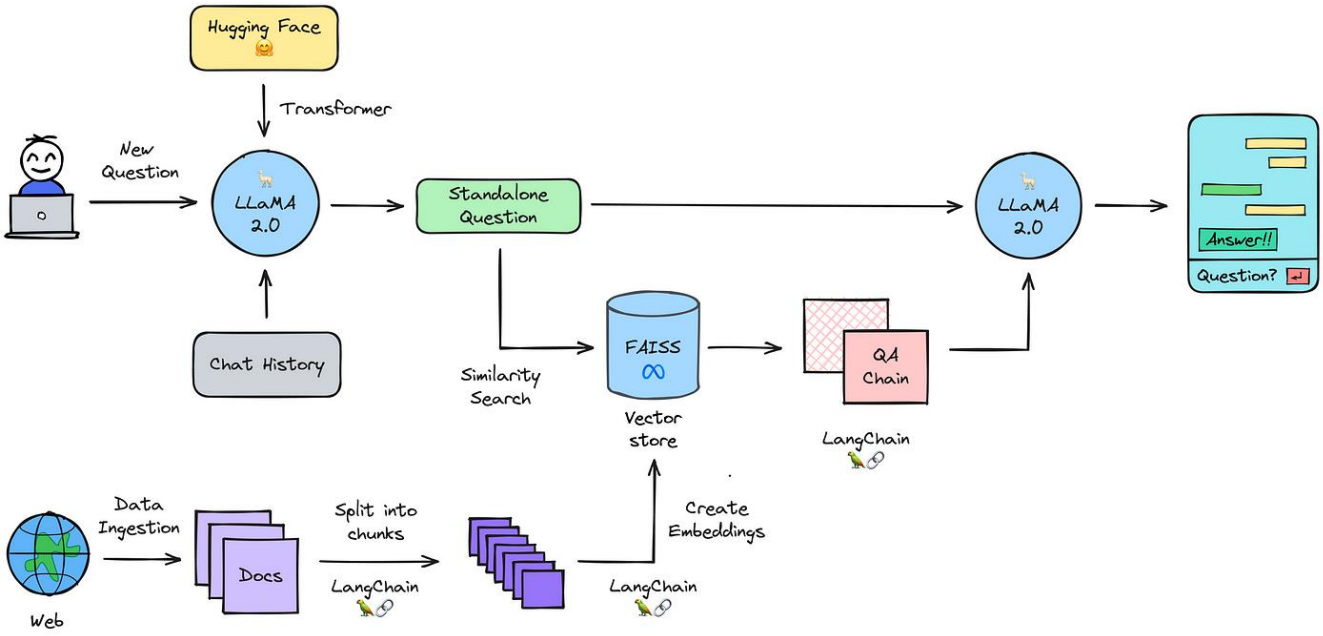
- **FAISS (Facebook AI Similarity Search)** is a library for efficient similarity search and clustering of dense vectors. It can search multimedia documents (e.g. images) in ways that are inefficient or impossible with standard database engines (SQL). It contains algorithms that search in sets of vectors of any size, up to ones that possibly do not fit in RAM. It also contains supporting code for evaluation and parameter tuning.



FAISS Indexing and Similarity Search — Source: engineering.fb.com

Process Flow

In this section, we will briefly describe each part of the process flow.



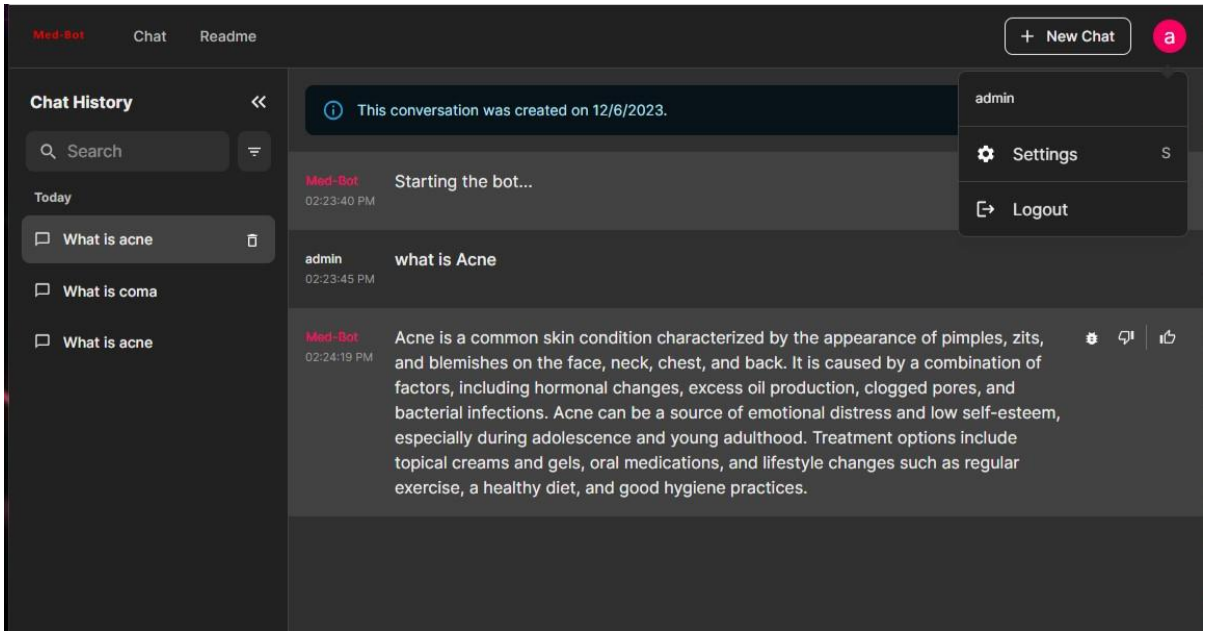
Process Flow Diagram

1. **Initialize model pipeline**: initializing text-generation pipeline with Hugging Face transformers for the pretrained Llama-2-7b-chat-hf model.
2. **Ingest data:** loading the data from the pdf in the form of text into the document loader.

3. **Split into chunks:** splitting the loaded text into smaller chunks. It is necessary to create small chunks of text because language models can handle limited amount of text.
4. **Create embeddings:** converting the chunks of text into numerical values, also known as embeddings. These embeddings are used to search and retrieve similar or relevant documents quickly in large databases, as they represent the semantic meaning of the text.
5. **Load embeddings into vector store:** loading the embeddings into a vector store i.e. “FAISS” in this case. Vector stores perform extremely well in similarity search using text embeddings compared to the traditional databases.
6. **Enable memory:** combing chat history with a new question and turn them into a single standalone question is quite important to enable the ability to ask follow up questions.
7. **Query data:** searching for the relevant information stored in vector store using the embeddings.
8. **Generate answer:** passing the standalone question and the relevant information to the question-answering chain where the language model is used to generate an answer.

EXPERIMENTAL SETUP

Our experimental setup, incorporating Meta's LLaMA 2 model (7B variant), LangChain, and the FAISS library, demonstrated robustness through real-world simulations. To enhance user interaction, we integrated a dynamic chat interface, exemplified in the accompanying screenshot, allowing users to seamlessly engage with the chatbot. This innovative feature complemented the efficiency of our question-answering chain, showcasing the adaptability and user-friendliness of our medical chatbot in delivering precise information for diverse healthcare queries.



RESULTS, DISCUSSION, AND OUTCOMES

Unveiling the Power of the Medical Chatbot

Our experiments demonstrated the chatbot's ability to provide accurate and contextually relevant answers to user queries. The system exhibited robustness against various input styles and proved to be a valuable resource for users seeking medical information.

Implications of Results on Healthcare

The successful implementation of our chatbot holds significant implications for the healthcare industry. Improved access to reliable medical information can empower users to make informed decisions about their health. Healthcare professionals can also benefit from the quick retrieval of relevant information, enhancing the efficiency of their work.

POTENTIAL DIRECTIONS FOR FUTURE WORK AND IMPROVEMENTS

Continuing the Innovation Journey

While our medical chatbot has shown promising results, there are several avenues for future improvement:

- **Expansion of Database**: Continuously updating and expanding the vector database to encompass the latest medical research.
- **User Interaction Enhancement**: Incorporating user feedback mechanisms to improve the chatbot's response based on real-world usage.
- **Multimodal Integration**: Exploring the integration of additional data modalities, such as images or voice, for a more comprehensive user experience.

CONCLUSION

In conclusion, our AI-powered medical chatbot represents a significant step forward in bridging the gap between users and medical information. By combining advanced NLP techniques, custom language models, and an efficient vector database, we've created a versatile tool with the potential to revolutionize healthcare communication. The positive outcomes from our experiments underscore the transformative impact AI can have in the healthcare domain. As we look ahead, the continuous refinement of our system and exploration of new possibilities promise a future where AI plays a pivotal role in shaping the way we access and understand medical information.

REFERENCES

[1] https://huggingface.co/blog/llama2

[2] https://venturebeat.com/ai/llama-2-how-to-access-and-use-metas-versatile-open-source-chatbot-right-now/

[3] https://www.pinecone.io/learn/series/langchain/langchain-intro/

[4] https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/

[5] https://ai.meta.com/tools/faiss/

[6] https://blog.bytebytego.com/p/how-to-build-a-smart-chatbot-in-10

[7] https://newsletter.theaiedge.io/p/deep-dive-building-a-smart-chatbot

[8] https://www.youtube.com/watch?v=6iHVJyX2e50

[9] https://github.com/pinecone-io/examples/blob/master/learn/generation/llm-field-guide/llama-2/llama-2-70b-chat-agent.ipynb