HEALPAL CHATMATE: AI DRIVEN DISEASE DIAGNOSIS AND RECOMMENDATION SYSTEM

***Abstract* - This research paper introduces an innovative healthcare paradigm by leveraging the LLM model Lamma 2 within the LangChain framework to develop a HealthBot capable of interpreting text-inputted symptoms and delivering personalized medication recommendations. The HealthBot's utilization of state-of-the-art natural language processing facilitates a user-friendly interaction, allowing individuals to articulate symptoms in natural language. The synergy between Lamma 2 and LangChain ensures efficient communication, enabling dynamic and responsive user experiences. The study delves into the technical intricacies, ethical considerations, and practical evaluations of the HealthBot, contributing valuable insights to the burgeoning field of AI-driven healthcare, particularly in the realms of disease diagnosis and medication recommendation systems.**

**Keywords:** LLM, Langchain, Lamma2, Prompt

1. **INTODUCTION**

In the era of rapid technological advancement, the integration of artificial intelligence (AI) into healthcare has emerged as a transformative force, offering innovative solutions to long-standing challenges. One such groundbreaking development is the advent of HealthBots, intelligent systems designed to assist individuals in assessing and understanding their health conditions based on reported symptoms. This research paper

delves into the realm of HealthBots, specifically focusing on their role in providing valuable insights into disease identification and offering general medication guidance.The traditional healthcare system often faces challenges such as limited accessibility, long waiting times, and escalating costs. HealthBots address these issues by leveraging AI algorithms to analyze symptom data input by users, facilitating a preliminary diagnosis and suggesting general medications associated with specific diseases. This technology not only empowers individuals to take an active role in managing their health but also contributes to the overall efficiency of healthcare delivery.This research paper undertakes a comprehensive exploration of HealPal ChatMate, delving into the innovative ways in which it harnesses AI to empower individuals in understanding and managing their health conditions. A significant focus of this research will be dedicated to evaluating HealPal ChatMate's recommendation system for disease management. By analyzing real-world scenarios and instances where the system has effectively recommended courses of action or medications, we aim to provide insights into its potential impact on personalized healthcare and patient outcomes.

1. **BACKGROUND** **INFORMATION**
2. *LLM (LARGE LANGUAGE MODEL)*

Large language models are sophisticated artificial intelligence systems that can comprehend and produce writing that resembles that of a person, such as GPT-3 (Generative Pre-trained Transformer 3). These models have been trained on a massive quantity of text data from the internet and are constructed using deep learning techniques. When there is little or no domain-specific data available for training, they are employed. The model's strong inductive bias and ability to produce meaningful representations from little or no input are key factors in both few-shot and zero-shot learning approaches, which are included in these situations.They are trained on massive datasets of text and code, which allows them to learn the patterns and rules of language. This enables them to perform a wide range of tasks, including:

* Text generation: LLMs can generate text that is similar to human-written text, including poems, code, scripts, musical pieces, email, letters, etc.
* Translation: LLMs can translate text between different languages.
* Question answering: LLMs can answer questions about a variety of topics.
* Summarization: LLMs can summarize text into a shorter and more concise form.
* Sentiment analysis: LLMs can identify the sentiment of a piece of text, such as whether it is positive, negative, or neutral.

1. *PROMPT ENGINEERING*

In the realm of artificial intelligence, prompt engineering is a crucial technique for effectively guiding large language models (LLMs) to generate high-quality outputs. It involves carefully crafting prompts, which are instructions or questions provided to LLMs, to steer them towards desired results. By meticulously designing prompts, prompt engineers can effectively control the style, tone, and content of the generated text, ensuring that the outputs align with specific goals or requirements.The process of prompt

engineering begins with a thorough understanding of the desired outcome. The prompt engineer must clearly define the task at hand and identify the specific characteristics of the desired output. This may involve considering aspects such as the level of formality, the target audience, or the desired tone of the generated text. Once the desired outcome is clearly defined, the prompt engineer begins constructing the prompt. This involves crafting a clear and concise instruction that conveys the task to the LLM in an unambiguous manner. The prompt should provide sufficient context and guidance without being overly prescriptive or restrictive. A well-designed prompt should effectively balance providing direction to the LLM while allowing it to exercise its creative abilities. It should strike a delicate balance between specificity and openness, ensuring that the generated output meets the desired criteria while also retaining a sense of originality and creativity. Prompt engineering plays a critical role in harnessing the power of LLMs to generate high-quality and relevant outputs. By carefully crafting prompts that provide clear instructions and guidance, prompt engineers can effectively control the style, tone, and content of the generated text, ensuring that LLMs are utilized to their full potential.

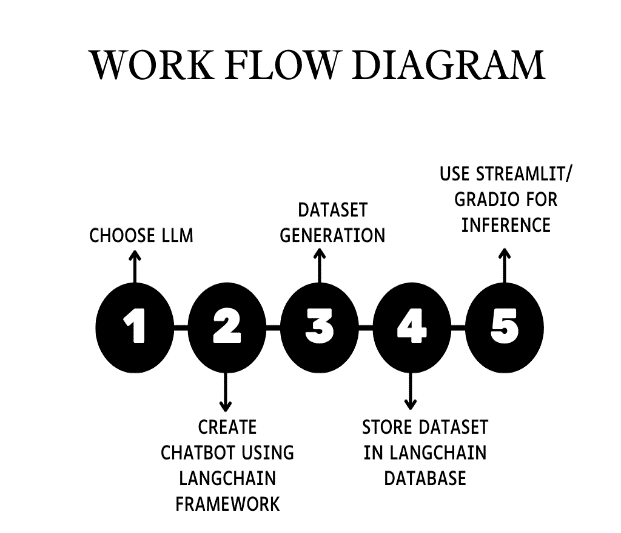
1. *QUANTIZATION*

Quantization is a technique used to reduce the size and complexity of large language models (LLMs). It is done by converting the weights and activations of an LLM from high-precision floating-point numbers to lower-precision integer numbers. Quantization can significantly reduce the memory footprint of the LLM and improve its inference performance. Quantization has several benefits for LLMs, including reduced model size, reduced inference latency, and reduced power consumption. Quantization involves reducing the precision of the model's parameters, typically from 32-bit floating-point numbers to lower bit representations, such as 16-bit or even 8-bit integers. This process aims to maintain the model's overall performance while significantly reducing the amount of memory needed to store the model and speeding up the computations during inference. By reducing the model size and computational requirements, quantization enables more efficient deployment without sacrificing model accuracy to a significant extent. This becomes crucial in real-world applications where resources like memory and processing power are limited.

2. **Literature** **review**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Work** | **Year** | **Author** | **Results** | **Future Scope** |
| [Llama2-Medical-Chatbot](https://github.com/AIAnytime/Llama2-Medical-Chatbot#llama2-medical-chatbot) | September 2023 | manjunathshiva | Sentence Transformers and Llama2 were used to create this medical bot. Langchain and Chainlit power the bot. | Can use other models for better results |
| Health-Bot (Healthcare Chatbot) | November 2021 | Sanket Muchhala, Avisha Jain, Sudhanshu Rai | NLP is employed for chatbots, and the Multinomial Naive Bayes algorithm is used for text classification. AI chatbot datasets (HealthBot): Dataset for disease and their symptoms | Looking forward to using AI in helping people detecting diseases- supported symptoms more accurately |
| Healthcare Chatbot using Natural Language Processing | April 2020 | Papiya Mahajan, Rinku Wankhade, Anup Jawade, Pragati Dange, Aishwarya Bhoge | The purpose of the application is to get a prompt response from the bot, meaning that it will provide the user the right answer right away. | The economical of chatbot will be improved by adding a lot of combination of words and increasing the use of database information so of the medical chatbot may handle all type of diseases |
| Intelligent Healthbot for Transforming Healthcare | April 2019 | Vivek Katariya, Prof. Vitthal S Gutte | The chatbot uses artificial intelligence and machine learning techniques to overcome the limitations of traditional human-machine interaction. This eliminates bias and enables the patient to communicate freely and naturally. | Will be working on adding more intents and better specification of entities to cover more symptoms and to make chatbot able to diagnose more diseases. |
| MedChatBot: An UMLS based Chatbot for Medical Students | October 2018 | Hameedullah Kazi, B.S. Chowdhry and Zeesha | Focuses on a medical chatbot design that utilizes AIML. The Chatter Bean AIML interpreter, which is based on Java, is used to build this Chatbot idea. | It intends to incorporate Medchatbot in the existing METEOR tutoring system for medical PBL, which will greatly facilitate natural dialog between the student users and the tutoring system |
| Artificial Intelligence for Chatbots in Mental Health: Opportunities and Challenges | August 2021 | Kerstin Denecke, Alaa Abd-Alrazaq, Mowafa Househ | The purpose of a chatbot system is to mimic human speech. The text-driven, uniform widgets, pictures, and unified style of chatbots make it simple to initiate conversation with one. | It shows different chatbots features and will continue getting better preferences |
| Healthcare chatbot using NLP and Flask | May 2022 | Kshitij Thakre, Dr. P. R. Rothe, Sakshi Kukade, Pranali Shinde, Komal Madame | A chatbot is a software program that converses with users either textually or audibly. | The economy of the chatbot will be improved by adding a lot of combinations of words and increasing the use of database information so the medical chatbot may handle all types of diseases. |
| Healthcare Chatbot using Natural Language Processing | 2021 | Tamrakar, Rohit & Wani, Niraj | The CHATBOT utilizes the concepts of Artificial Intelligence and Machine Learning to interact with people virtually. Firstly, the development history is reviewed, followed by an explanation of the architecture, and different CHATBOT classifications | The implementation of personalized drugs would with success save several lives and build a medical awareness among the people. |

1. **WORK FLOW DIAGRAM**

****

1. **METHODOLOGY**
2. *LLM MODEL : LAMMA 2*

Developed by Meta AI, Llama 2 is a transformer-based model with 7 billion parameters, trained on a massive dataset of text and code. This extensive training enables the model to grasp intricate linguistic patterns and nuances, empowering it to excel in various NLP applications. At the heart of Llama 2 lies the transformer architecture, a deep learning model specifically designed for NLP tasks. The transformer employs a technique called self-attention, which allows the model to focus on the most relevant portions of the input text while generating its response. This mechanism enables Llama 2 to produce coherent and relevant outputs, even when dealing with complex or ambiguous language.The transformer architecture consists of two primary components: the encoder and the decoder. The encoder processes the input text, extracting key information and establishing relationships between words. The decoder, on the other hand, generates the output text, utilizing the encoded representation of the input and the desired task specifications.

Working of lamma 2:

1. The input text is first tokenized into a sequence of words. This means that the text is broken down into individual words, and each word is assigned a unique numerical ID.
2. The model then generates embeddings for each word. Embeddings are vector representations of words that capture their meaning. The embeddings are generated by using a neural network that is trained on a large corpus of text data.
3. The embeddings of the input words are then passed through the transformer encoder. The transformer encoder is a multi-layer neural network that uses self-attention to generate a representation of the input text. The encoder takes into account the relationships between the words in the input text, and it learns to focus on the most relevant parts of the text.
4. The output of the transformer encoder is then passed through the transformer decoder. The transformer decoder is another multi-layer neural network that uses self-attention to generate the output of the model. The decoder takes into account the input text and the desired output, and it learns to generate text that is relevant to the input and that is grammatically correct.
5. The output of the transformer decoder is then processed to generate the final output of the model. This may involve using a language model to generate text that is more fluent and natural-sounding.
6. *LANGCHAIN FRAMEWORK*

LangChain emerges as a groundbreaking framework that revolutionizes the development and deployment of applications powered by large language models (LLMs). This innovative platform provides a standardized and streamlined approach to connecting LLMs with other essential components, such as data sources and user interfaces. By eliminating the complexities of integrating these diverse elements, LangChain empowers developers to create sophisticated LLM-driven applications with greater efficiency and enhanced capabilities.At the core of LangChain's functionality lies its chaining mechanism, which enables the seamless interconnection of various components into a cohesive workflow. This modular approach allows developers to assemble chains tailored to specific application requirements, ensuring that the LLM's capabilities are harnessed effectively. By combining the LLM's text generation prowess with data sources that provide contextual information and user interfaces that facilitate interactive user experiences, LangChain empowers developers to create intelligent and engaging applications.LangChain's comprehensive features extend beyond its chaining mechanism, offering a range of tools and functionalities that enhance the development process. The framework provides a robust runtime environment for executing chains and agents, enabling developers to test and debug their applications seamlessly. Additionally, LangChain offers a library of pre-built chains and agents, providing developers with readily available modules for common tasks, such as sentiment analysis and question answering. This collection of reusable components accelerates development and ensures consistent quality across applications.

Steps of chatbot creation:

1. Prompt Template

A prompt template is a reusable component that provides a structured way to generate prompts for large language models

(LLMs).Prompt templates can be used to generate prompts for a variety of tasks, such as text generation, translation, and question answering.

LangChain provides a number of pre-built prompt templates, such as:

* Text Generation: This prompt template can be used to generate a variety of creative text formats, such as poems, code, scripts, musical pieces, email, letters, etc.
* Translation: This prompt template can be used to translate text between different languages.
* Question Answering: This prompt template can be used to answer questions about a variety of topics.

You can also create your own custom prompt templates. To do this, you will need to define the input variables for the prompt template and the template itself.

Here is an example of a custom prompt template for generating poems:

{

"inputVariables": ["adjective"],

"template": "Tell me a {adjective} poem."

}

This prompt template will generate a prompt that asks the LLM to generate a poem with the specified adjective. For example, if you provide the value "funny" for the adjective variable, the prompt will be:

Tell me a funny poem.

1. Conversational Buffer Conversational buffers are an essential feature of LangChain, contributing significantly to the coherence and context-awareness of LLM-powered conversations. These temporary storage spaces hold the history of interactions between users and LLMs, allowing the models to access previous statements and responses for more informed and relevant outputs.Conversational buffers provide LLMs with access to the conversation history, enabling them to better grasp the context of current prompts and questions. This results in more pertinent and consistent responses that align with the overall conversation flow. The ability to access past interactions allows LLMs to emulate natural human conversations, making interactions more engaging and personalized. They can refer back to previous statements, address specific topics raised earlier, and maintain a cohesive thread throughout the conversation.By storing past interactions, conversational buffers minimize the need for users to repeat themselves or rephrase questions. LLMs can access this information and provide responses that build upon the existing context, avoiding unnecessary repetition.
2. LLM Chain

An LLM chain in LangChain is a fundamental component that facilitates the interaction between users and large language models (LLMs). It serves as a structured workflow that connects different components, each playing a specific role in processing user inputs and generating LLM outputs. Chains are composed of reusable components, allowing for flexibility in application development and easy integration with other systems.By providing structured guidance to the LLM, chains help ensure that generated outputs are relevant, coherent, and aligned with user expectations.Chains simplify the development process by providing a

standardized framework for interacting with LLMs, reducing the need for complex coding.Chains can be adapted to a wide range of tasks, making them a versatile tool for building LLM-powered applications in various domains.LLM chains in LangChain play a crucial role in connecting users with the power of LLMs, enabling the development of intelligent and engaging applications that leverage the capabilities of these powerful language models.

1. **Dataset Generation**

Two types of datasets are used in this project some were existing datasets and some were created with the information already present and with medical help.

1. **DATABASE: FAISS CPU**

FAISS CPU is a vector store component within the LangChain framework that utilizes the FAISS library for efficient similarity search and clustering of dense vectors on CPU-based systems. It is specifically optimized for CPU-based computation, making it a suitable choice for scenarios where GPU resources are unavailable or not preferred. It is designed for rapid identification of similar vectors within large datasets, enabling efficient search and retrieval operations.It can handle large volumes of data, making it suitable for applications with extensive vector storage and retrieval requirements. Optimized for CPU-based computation, FAISS CPU can effectively utilize available CPU resources, making it a cost-effective solution for many scenarios. It seamlessly integrates with the LangChain framework, enabling developers to easily incorporate vector storage and retrieval capabilities into their LLM-powered applications.

1. **Retrieval Question Answering (QA)**

Retrieval Question Answering (QA) is a subfield of natural language processing (NLP) that focuses on retrieving relevant answers to questions from large text corpora. It involves understanding the intent of the question and identifying the most relevant passages of text that contain the answer. Retrieval QA systems are typically faster and more accurate, and they are able to handle a wider range of questions.They can search large text corpora in a matter of milliseconds, making them ideal for real-time applications.They are typically more accurate than generative QA systems, as they are able to leverage the existing information in the text corpus.They can handle a wider range of questions than generative QA systems, including questions that require reasoning and inference.Retrieval QA systems typically consist of two main components:

1. **A question understanding module:** This module is responsible for extracting the key information from the question, such as the topic, entities, and keywords.
2. **A retrieval module:** This module is responsible for searching the text corpus for passages that are relevant to the question. The retrieved passages are then ranked based on their relevance to the question, and the top-ranked passages are returned as the answer.
3. **RESULTS**

Inference time is around 15 – 20 seconds .It is giving composition of medicines to be taken as an instant relievant for given symptoms it also suggests name of the medicine . It also suggest more than one solution to any problem which gives more broad options to a patient to search for medicines as different regions may

have different medine names for same composition.

1. **FUTURE SCOPE**

We will try to reduce our inference time so that we can get results earlier. We will also try other models for better results. Further, we will also work on taking image based inputs which will help patients to be more clear about their symptoms for some skin diseases or infections which specifically need different types of treatment or eye problems. We will also try to add a database through which we can suggest doctors that are present in a nearby location.

1. **REFERENCE**

[1] Jain, Avisha & Muchhala, Sanket. (2020). Health-Bot (Healthcare Chatbot).

[2] Rarhi, Krishnendu & Bhattacharya, Abhishek & Mishra, Abhishek & Mandal, Krishnasis. (2017). Automated Medical Chatbot.

SSRN Electronic Journal. 10.2139/ssrn.3090881.

[3] Katariya, Vivek & Vitthal, Shinde & Gutte, Vitthal & Devare, Manoj. (2019). Intelligent Healthbot for Transforming Healthcare.

Tamrakar, Rohit & Wani, Niraj. (2021). Design and Development of CHATBOT: A Review.

[4] Denecke, Kerstin & Abd-alrazaq, Alaa & Househ, Mowafa. (2021). Artificial Intelligence for Chatbots in Mental Health: Opportunities and Challenges. 10.1007/978-3-030-67303-1\_10.

[5] Gupta, Aishwarya. (2020). Introduction to AI Chatbots. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS070143.

[6] Reyner, Andrew & Tjiptomongsoguno, Wibowo & Chen, Audrey & Sanyoto, Hubert & Irwansyah, Edy & Kanigoro, Bayu. (2020). Medical Chatbot Techniques: A Review. 10.1007/978-3-030-63322-6\_28.

[7]<https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/>

[8] <https://docs.langchain.com/docs/>

[9] <https://ai.meta.com/llama/>

[10]<https://dataconomy.com/2023/07/19/meta-ai-what-is-llama-2-and-how-to-use/>

[11]https://analyzingalpha.com/langchain-python-tutorial#Memory