**Building RAG Chatbot with Langchain**

**A diagram of a diagram of a diagram

Description automatically generated**

**Introduction**

In this project, we aimed to build a knowledge base using the Llama 2 dataset sourced from ArXiv papers. Llama 2 comprises chunks of text from academic papers, and our goal was to transform this dataset into a searchable knowledge base using vector embeddings and Pinecone, a vector database service.

**Dataset Overview**

The dataset sourced from the Llama 2 ArXiv papers consists of chunks of text from academic papers available on ArXiv, a repository of electronic preprints. Each entry in the dataset represents a segment of text extracted from these papers.

A screenshot of a computer

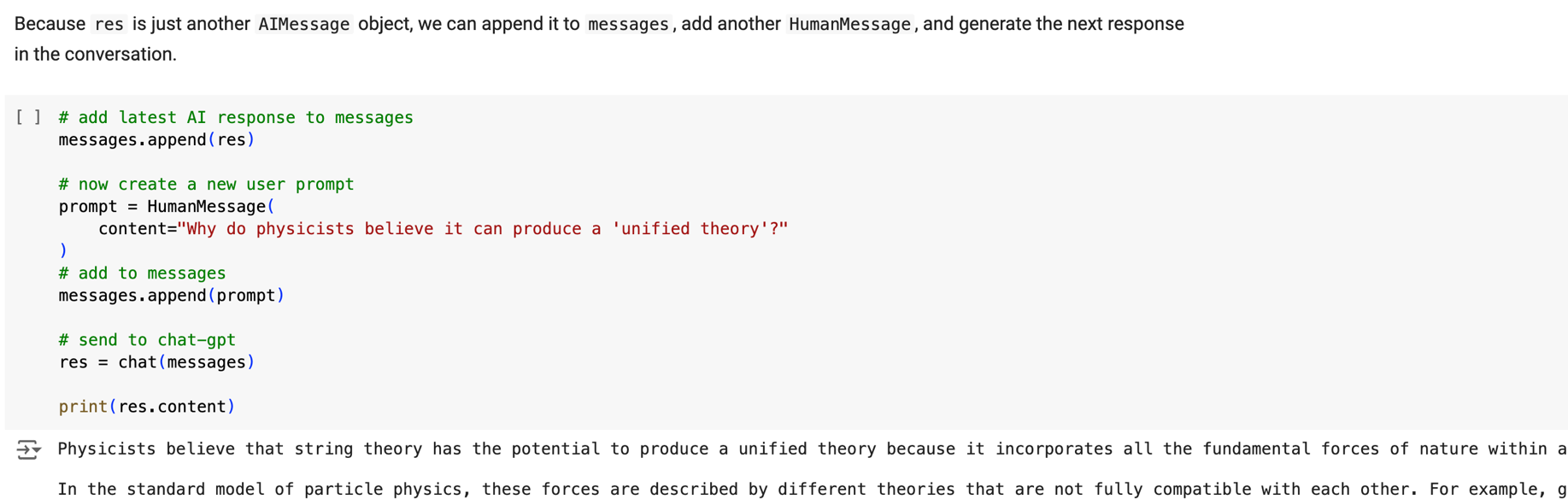
Description automatically generated

**Building a Chatbot (no RAG)**

In this section, we utilized the LangChain library to construct a chatbot without retrieval augmentation. The ChatOpenAI object from LangChain facilitated this process, leveraging OpenAI's GPT-3.5-turbo model for generating responses.

**Key Steps:**

1. **Initialization and Setup:**
   * We set up the environment by initializing the ChatOpenAI object with our OpenAI API key.
   * Messages were structured using LangChain's schema, distinguishing between SystemMessage (bot-initiated), HumanMessage (user-initiated), and AIMessage (AI-generated) components.
2. **Dialogue Flow:**
   * The conversation flow was managed by passing a sequence of messages to the ChatOpenAI object.
   * Each user query triggered a response from the GPT-3.5-turbo model, providing detailed explanations and insights.
3. **Example Interaction:**
   * An example included the user query about understanding string theory, which prompted the chatbot to deliver a comprehensive overview.
   * The AI response elaborated on string theory's fundamentals, including particle nature, extra dimensions, unification of forces, multiverse concepts, ongoing research challenges, and required background knowledge.
4. **Continued Dialogue:**
   * Subsequent user questions could be seamlessly integrated into the ongoing conversation by appending new HumanMessage prompts and generating corresponding AIMessage responses.

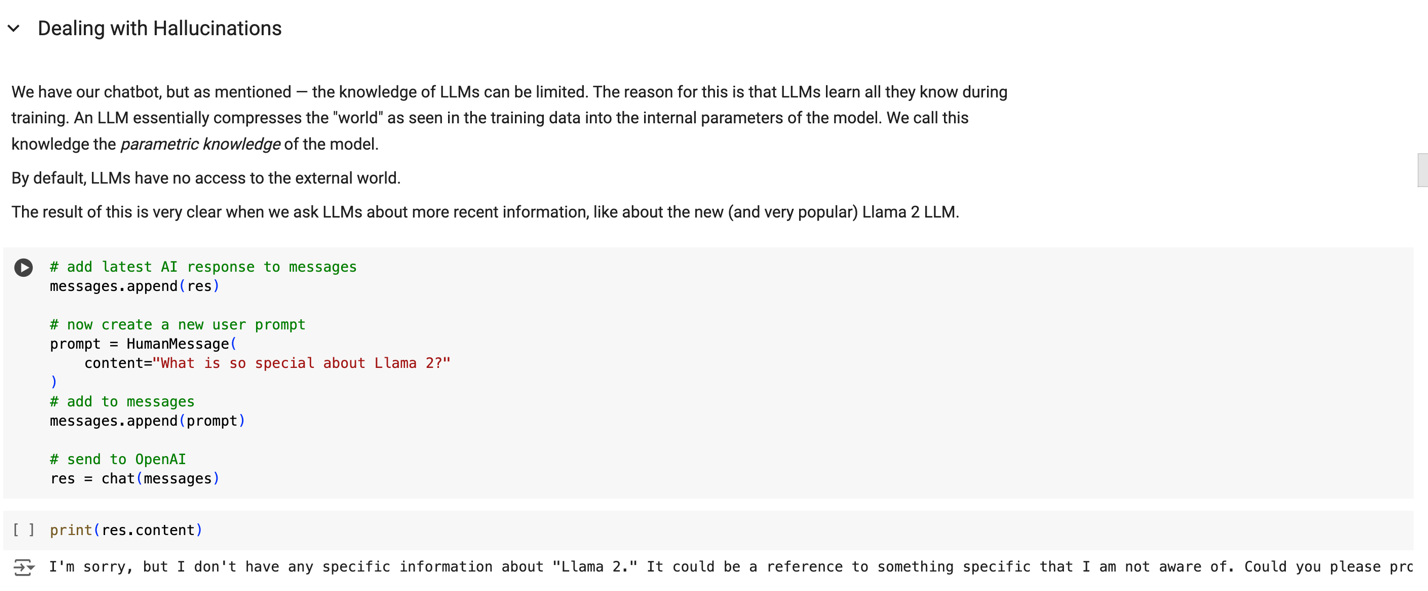


**Dealing with Hallucinations**

The section addresses the limitations of Language Model Models (LLMs), such as GPT-3.5-turbo, due to their reliance on parametric knowledge acquired during training. LLMs lack access to real-time or specific external information, which becomes evident when queried about recent developments like the Llama 2 LLM.

**Key Points:**

1. **Parametric Knowledge:**
   * LLMs encapsulate knowledge from training data into their internal parameters. They are unable to access or incorporate new, external information not present during training.
2. **Hallucinations and Limitations:**
   * LLMs may respond to queries about unfamiliar topics with potentially misleading or incorrect information, termed as hallucinations. This occurs when the model generates a plausible-sounding response without factual basis.
3. **Source Knowledge Approach:**
   * To mitigate these issues, the concept of source knowledge is introduced. Source knowledge involves feeding specific information into the model via the query prompt. This external knowledge supplements the model's internal understanding, enhancing the accuracy and relevance of responses.
4. **Implementation Example:**
   * An example demonstrates augmenting a query about "LLMChain in LangChain" with predefined information about LangChain from its documentation. This enriched prompt instructs the LLM on how to incorporate external knowledge to improve response quality.
5. **Future Applications:**
   * The potential of integrating external databases, such as vector databases like Pinecone, is highlighted. These databases could provide real-time, contextually relevant information to enhance LLM responses further.



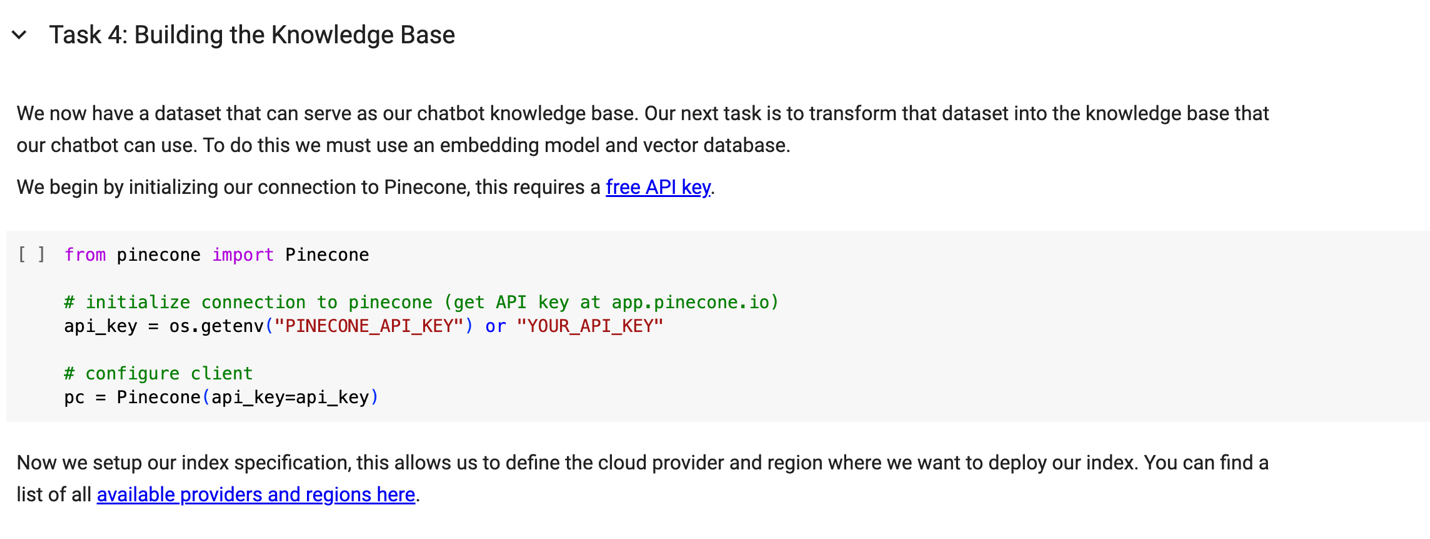
**Importing Data from Hugging Face Datasets**

This part focused on practical steps to import external data using the Hugging Face Datasets library. Specifically, we utilized the "jamescalam/llama-2-arxiv-papers" dataset, sourced from ArXiv papers. This dataset serves as a foundational knowledge base for our chatbot, containing scholarly articles relevant to our domain of interest.



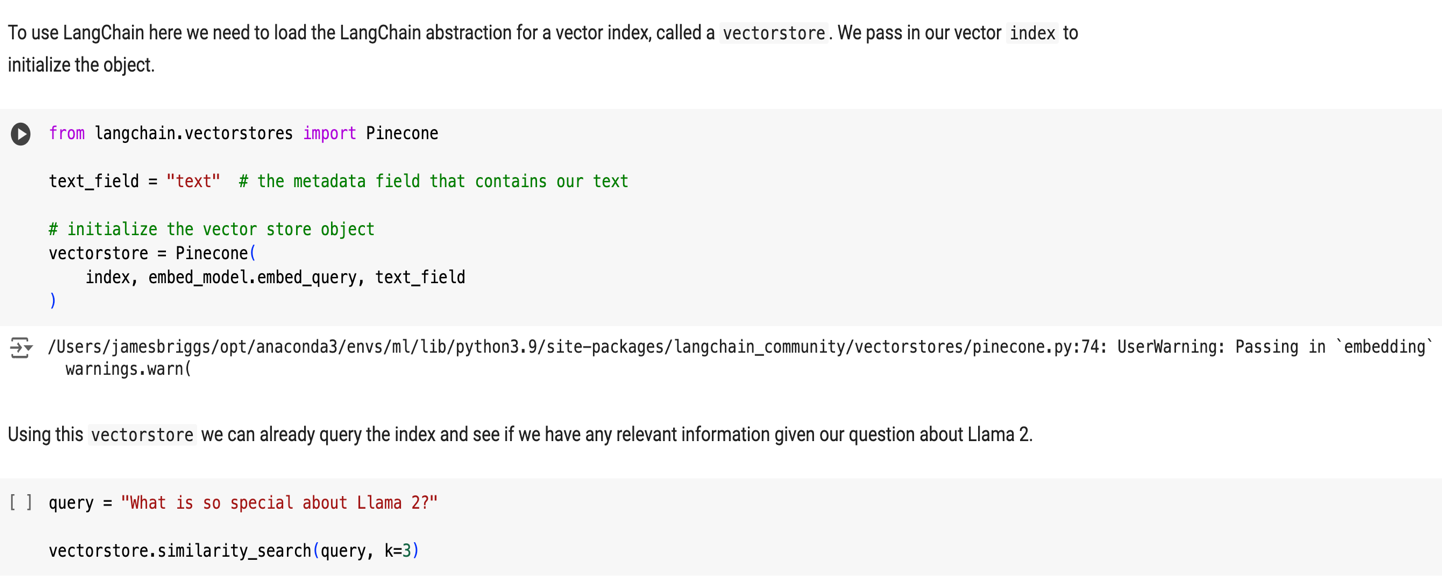
**Building the Knowledge Base with Pinecone**

Here, we detailed the process of transforming the imported dataset into a functional knowledge base using Pinecone, a vector database service. We configured and initialized an index to store vector embeddings derived from the dataset, enabling efficient storage and retrieval of information critical for our chatbot's responses.



**Integrating Knowledge Base with Chatbot via LangChain**

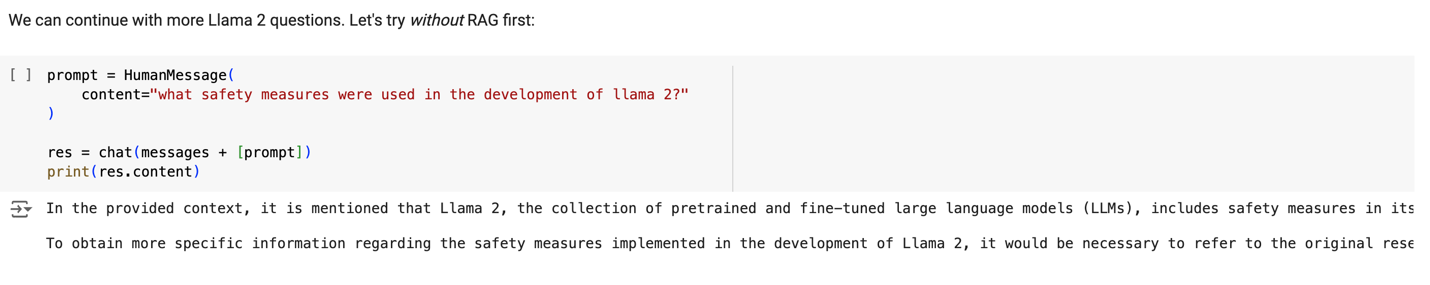
This section focused on integrating the vector index, built with Pinecone, into our chatbot framework using LangChain. By utilizing LangChain's vectorstore abstraction, we facilitated seamless interaction between the chatbot and the external knowledge base. This integration enhances the chatbot's capability to dynamically access and incorporate relevant information into its responses.



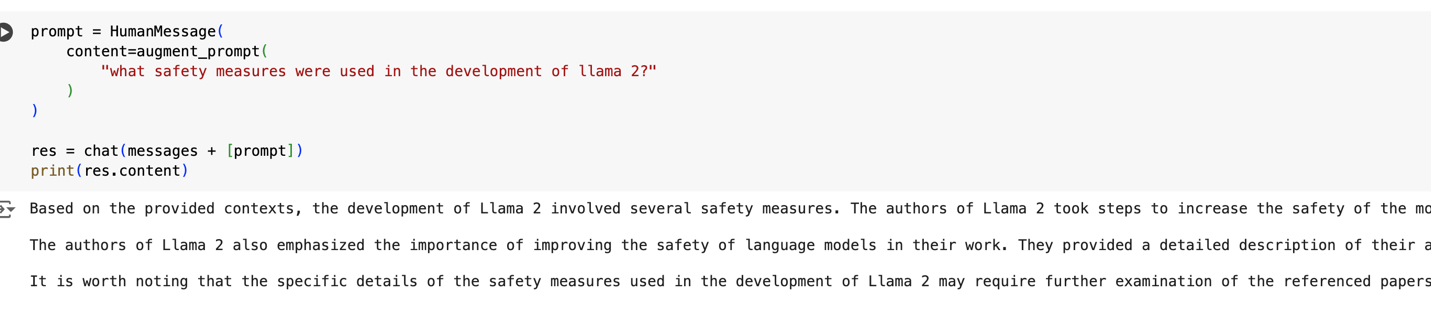
**Enhancing Response Quality with External Knowledge**

Demonstrating the practical application of augmented queries, this part illustrated how integrating external knowledge enriches the chatbot's responses. Using specific examples like queries about "Llama 2", we showcased how the chatbot can provide more nuanced and accurate information beyond its internal parametric knowledge, thereby improving overall response quality and user satisfaction.

**Response from the chatbot without RAG**



**Response from the chatbot with RAG**



**Conclusion and Future Directions**

In conclusion, leveraging external knowledge through augmented queries represents a significant advancement in enhancing conversational AI systems. This methodology not only addresses the limitations of LLMs but also opens avenues for further improvements and refinements in chatbot capabilities. Future considerations include expanding the scope of external datasets, refining integration processes, and exploring additional applications to enhance user interactions and satisfaction.

**Curated by SHARMI DAS**