Level 1:- Build a RAG (Retrieval-Augmented Generation) based system using any vector db, that empowers users to input semantic queries about the hotels they are searching for. The system should not only retrieve relevant hotels in the corresponding city but also utilize a decoder model to explain why a particular hotel matches their preferences.

Our solution we have used the Langchain as our framework, FIASS as our vector database

And we have used OpenAI API for embeddings and AzureChatOpenAI(gpt-4) as our large language model.

Steps:

1. We started by cleaning the dataset, the first thing we noticed were there were links of images in a column so we removed that column since they are not much use even if we had it or not.
2. We also noticed there were lot of missing values in the hotel descriptions but one good thing was there were already descriptions for the same hotel hence we autofill the descriptions of the hotels respectively.
3. Then we started working on the price, We made 3 categories out of that price\_range column i.e ($-low, $$-medium and $$$-high price respectively)
4. After some cleaning we had a good dataset, Firstly we converted the css dataset into JSON data because it is more structured and we thought it would be easy for the us to store them in a vector database and retrieve the results.
5. We have used the RetrievalQA chain for the level one because it was mostly just asking the question getting an answer.
6. Lastly, we have used gpt-4 as our decoder and we used few-shot-techniques to let the model understand what kind of question must be answered and what not.

Level-2: - Augment the Level 1 RAG model by integrating [Traversaal.ai](https://traversaal.ai/)'s Ares API, which performs real-time internet searches. Participants are encouraged to enhance their RAG applications by incorporating relevant details about hotels or locations obtained dynamically through the Ares API. E.g. *“food near these hotels”, “things to do in this area” or “articles/blogs about the hotel not available in the dataset”*

Participants can utilize this api endpoint by signing up at: [api.traversaal.ai](https://api.traversaal.ai/) and get access to 100 web searches for free per user - no credit card needed

You can see Ares documentation [here](https://docs.traversaal.ai/introduction).

In this part of our problem we have used traversal.ai’s Ares API as mentioned and similar to level 1 we have used the same dataset. The main aim of the problem is to get the real time data for food places, hotels, cost of hotels etc.

A diagram of a process

Description automatically generated

Figure 1.

Steps:

1. Since we already had our FIASS database stored we didn’t need any other vector database.
2. We made a very simple function that takes the query and gets the response from the API and the same query is used to query on the vector database as well.
3. Now we have integrated both RAG and API calls as shown in the below Figure 1.
4. After getting the response from both the RAG and API we now given it to our LLM along with query, to get the final result.

Level-3: - Develop a conversational style chatbot capable of engaging in multiple conversations with users about their hotel preferences. The chatbot should seamlessly invoke OpenAI functions to generate RAG outputs. Additionally, participants are expected to leverage the Ares API endpoint within the chatbot to provide users with real-time information.

Steps: - I wouldn’t say this is hardest but yes definitely was fun solving compared to the other levels.

1. This is a chatbot designed to assist users with hotel-related inquiries. Users can input their questions or requests, and the chatbot will provide helpful responses.
2. The main aim of this problem was to keep the human as engaging as possible and let the bot ask the questions and then take an action.
3. Unlike previous levels in this we had to make tools out of our functions such that agent can decide when it must call the Ares\_API or the RAG function.

How to Use(https://huggingface.co/spaces/Siddartha10/traversaal\_ai\_conversational\_bot)

User Input: Enter your question or request in the "User Input" text box.

Submit: Click the "Submit" button to send your input to the chatbot.

Chatbot Response: The chatbot will process your input and provide a response. You can view the response in the "Chatbot Response" text area below.

Exit: If you're done using the chatbot, you can click the "Exit" button to close the application.

Features:

Real-time Internet Searches: The chatbot can perform real-time internet searches using the Traversaal AI Ares API to fetch relevant information.

Question Answering: It uses a powerful RAG (Retrieval-Augmented Generation) model to answer questions based on the provided context.

Customizable Responses: The chatbot's responses are customizable and tailored to provide informative and personalized explanations.

Note: Ensure to click the "Submit" button after entering your question or request to receive a response from the chatbot.