Case Study: Generative AI Powered Agent Assist BoT for a BPO

Client Overview:

The client is a leading BPO in India which support multiple large banks as their client. They want to provide knowledge management solution for the Agent to effectively respond to customer queries

Business Challenge:

Recognizing the limitations of traditional chat systems in facilitating context-aware conversations, our client sought a solution that would allow users to engage in meaningful and relevant interactions based on specific topics. Our mission was to bridge this gap by providing a platform where users could seamlessly create and choose knowledge bases aligned with their preferences or needs.

Solution: Retrieval Augmented Generation in Langchain Framework:

This innovative solution empowers users to create and engage in contextually rich conversations by selecting specific knowledge bases tailored to their interests. Leveraging advanced technologies like persistent vector stores, OpenAI embeddings, retrieval QA chains, and language models, we've delivered a unique conversational AI experience.

1. System Architecture:

- **Langchain Framework Implementation:** Utilized the Langchain framework to seamlessly integrate Retrieval Augmented Generation into our solution, ensuring a robust and scalable architecture.

- **Vector Store and Embeddings:** Developed multiple vector stores or knowledge bases, each dedicated to a specific topic, utilizing OpenAI embeddings to represent document information effectively.

- **Retrieval QA Chain Enhancement:** Implemented a retrieval QA chain to enhance the accuracy of knowledge retrieval, enabling the system to respond effectively to user queries.

- **Language Model (LLM) Integration:** Integrated a language model (LLM) into the solution to generate context-aware and relevant responses to user queries, employing prompt engineering skills for optimal performance.

- **Evaluation:** Ragas framework assesses Retrieval Augmented Generation (RAG) and Retrieval pipelines. RAG enhances Large Language Models (LLM) using external data. Ragas evaluates metrics like faithfulness, answer relevancy, context precision, and context recall for a thorough assessment of RAG pipelines.

2. Components Utilized:

- **OpenAI Embeddings:** OpenAI embeddings contribute to the natural language understanding component, allowing the system to grasp the nuances of user queries.
- **ChromaDB:** A persistent ChromaDB is employed to store contextual information, enabling the system to understand and respond intelligently to user queries.
- **Retriever Object:** A dedicated retriever object is employed to efficiently retrieve relevant data from the Vector Database based on the similarity with the query provided by the user.
- **Prompt Template:** Prompt templates are predefined structures or formats for presenting user queries to the retrieval model. These templates are designed to guide the system on how to construct a retrieval-friendly query.
- **Advanced Language Model:** A robust language model, processes user queries, retrieved documents, and additional context. It produces detailed, contextually relevant responses that seamlessly align with the conversation's flow and user expectations.

Outcome: Elevated Conversational Experience:

- **Topic-Specific Conversations:** Users can now select knowledge bases aligned with their interests, facilitating conversations that are contextually relevant and personalized.
- **Context-Aware Responses:** The retrieval QA chain and language model work seamlessly to provide context-aware and accurate responses, enhancing the overall conversational experience.

- **Efficient Knowledge Retrieval:** Our solution ensures efficient knowledge retrieval by incorporating persistent vector stores, resulting in a seamless and user-friendly conversational interface.

Technology Stack:

- Langchain Framework: For overall system architecture and integration.
- OpenAI Embeddings: Ensuring effective representation of document information.
- Persistent Vector Stores: Storing and retrieving document embeddings.
- Retrieval QA Chain: Enhancing knowledge retrieval accuracy.
- Language Model (LLM): Generating context-aware responses.

Conclusion: Elevating Conversational AI Experiences:

Our company has successfully developed and implemented a cutting-edge solution that transforms traditional chat interactions into context-aware and personalized conversations. By leveraging the power of Retrieval Augmented Generation within the Langchain framework, we have provided our client with a powerful tool that enhances user engagement, knowledge retrieval, and overall conversational experiences. This case study stands as a testament to our commitment to pushing the boundaries of AI innovation and delivering solutions that redefine the landscape of conversational AI.