**Telecom multi-agent chat bot system**

**Implementation Document:**

The complete telecom multi-agent ai chat bot is built using langgraph and langchain along with Gemini 1.5 Flash as an LLM.

The System is divided into 2 agents:

1. Technical Agent: Which process the technical documents and answer to a user query which they are facing a problem.
2. Marketing Agent: Answers to a query related to advertisement dales data.

In-depth implementation of the Technical Agent:

Steps involved in building Technical Agent:

1. First converts the documents in PDF or DOCX format to text file and saves it.
2. It then converts the extracted text into chunks for embedding generation. We are using paragraph chunking for better embedding generation.
3. It then passes the text document to a Sentence Transformer `BAAI/ bge-large-en-v1.5`.
4. Once the text embeddings are generated, it then stores it into a vector database (Qdrant) for the retrieval of the text of user query. (For installation of qdrant vector db in a machine, please see user guide and maintenance document)
5. User query is also converted to embeddings.
6. Once the text embeddings are stored in vector database, it then uses cosine similarity to search for the query response from the vector db.
7. The resultant text is then passed to Gemini 1.5 Flash LLM with system prompt to generate the most accurate response of the user query.
8. Lastly it then returns the response from the LLM to the user.

In-depth implementation of the Marketing Agent:

This part requires installation of SQLite DB in the machine (Explained in user guide and maintenance document).

1. First the connection to SQLite DB is set-up and a table is created in an SQLite DB.
2. Once the table is created, it then stores the CSV or XLSX file to SQLite table.
3. An extensive system prompt is provided along with user query to generate the response of the user query. The prompt basically tells the LLM to convert the user query to an SQL query and then execute the SQL query into SQLite db and generate the output of the SQL query.
4. The entire prompt is then passed to Gemini 1.5 Flash LLM to generate the query and then this query is finally executed in SQLite DB to generate the output in a dataframe format.
5. The final output is then sent back to user.

We have also built intent recognition system to identify where the user query should be routed to (either marketing or technical agent).

Below are the steps implemented to build the intent recognition module.

1. Imported technical and marketing agent tools in this module.
2. Built agents out of these tools.
3. Built a supervisor agent to decide which tool needs to be called based on the user query. For this, we have written system prompt to route the user query to an appropriate agent (marketing or technical)
4. Once it is decided where to route the user query, it then pass the query to the respective agent and calls the main function which generates the appropriate response from LLM.
5. Workflow is created to maintain the state of the agent.
6. Once the response from the respective agent is generated, it then sends the response to a main supervisor agent and then finally sent back to the user.

Once all of the above 3 agents are built, we expose the supervisor agent to the user. User will enter the query related to marketing or technical or in general, the supervisor then calls the respective agent and then finally sent back the appropriate response to the user in the form of natural language.