Corpus Chat Bot Project Report

Overall Approach

The goal of this project was to create a chatbot that can answer customer queries based on a provided corpus about a wine business. The chatbot should respond to in-corpus questions and direct users to contact the business for out-of-corpus queries. The project was implemented using Dialog-flow for natural language understanding and a custom backend for dynamic responses and context handling.

Steps Taken

1. **Set Up Dialog-flow Project**: Created a new agent on Dialog-flow and configured it to handle wine-related queries.
2. **Extract Information from Corpus**: Parsed the provided Corpus.pdf and Sample Question Answers. json to identify key pieces of information.
3. **Create Intents and Entities**: Defined intents and entities in Dialog-flow based on the extracted information.
4. **Develop Fulfilment**: Implemented a webhook to handle complex queries and maintain conversation context.
5. **Handle Context and Follow-Up Intents**: Used Dialog-flow contexts to manage follow-up questions.
6. **Deploy and Test**: Created a minimalistic UI, integrated the chatbot with the website, and tested thoroughly.

frameworks/Libraries/Tools Used

Dialog-flow

- **Purpose**: Natural language understanding and managing intents/entities.
- **Usage**: Created intents for various queries (e.g., wine details, tasting room info) and entities for key elements (e.g., wine types, winemakers).

Node.js and Express

- **Purpose**: Backend server to handle webhook requests from Dialog-flow.
- **Usage**: Implemented the webhook endpoint to process Dialog-flow requests and return appropriate responses.

HTML/CSS

- **Purpose**: Frontend for the chatbot UI.
- **Usage**: Created a simple HTML page with a Dialogflow Messenger widget for user interaction.

Problems Faced

- Maintaining Conversation Context

The chatbot needed to handle follow-up questions and maintain context across multiple user queries.

Solution - Utilized Dialog-flow contexts effectively. Set output contexts in initial intents and used input contexts in follow-up intents to maintain the state of the conversation.

Conclusion

By following the outlined approach and utilizing Dialog-flow, Node.js, and a simple frontend UI, we successfully created a chatbot that meets the project requirements. The chatbot can answer questions based on the provided corpus and direct users to contact the business for out-of-corpus queries.