
Jessup Cellars Chatbot

Overview

This repository contains a Flask-based chatbot application developed for Jessup Cellars. The chatbot is designed to assist users with wine-related queries and supports multiple languages. It includes audio feedback functionality, which can be valuable for lead generation.

Technologies Used

- **Python**: The primary programming language used for development.
- **Flask**: A lightweight web framework used to create the chatbot server.
- **Transformers**: Used for natural language processing, specifically for question answering.
- **gTTS (Google Text-to-Speech)**: Converts chatbot text responses into spoken audio.
- **FuzzyWuzzy**: Provides fuzzy string matching to improve question-answer matching.
- **JSON**: Utilized for storing question-answer pairs in the dataset.

Features

- **Audio Feedback**: Responses are converted to audio using gTTS (Google Text-to-Speech), which helps in engaging users and tracking interactions. This audio feedback can be utilized for generating leads and analyzing user behavior.
- **Question Answering**: Utilizes a pre-trained question-answering (QA) model from the transformers library to provide accurate answers based on the provided corpus of wine-related information.
- **Contextual Follow-Up**: Maintains conversation history to handle follow-up questions effectively. The chatbot refers back to previous interactions to provide contextually relevant answers.
- **Customizable Responses**: The chatbot can be adapted to handle different types of user queries by modifying the JSON corpus.

Future Enhancements

- **Multilingual Support**: The chatbot can handle queries in multiple languages, enhancing accessibility for users from diverse linguistic backgrounds.
- **Expanded Language Support**: Add more languages and improve translation capabilities to cater to a broader audience.
- **Enhanced Lead Generation**: Develop more sophisticated methods for lead generation based on user interactions and preferences.

- **Predefined Questions**: Incorporate a library of predefined questions to provide users with instant answers to common queries.

Setup and Installation

To set up and run the application, follow these steps:

1. **Clone the Repository**

   ```
   git clone <repository_link>
   cd <repository_name>
   ```

2. **Create a Virtual Environment**

   ```
   python -m venv venv
   source venv/bin/activate  # On Windows use: venv\Scripts\activate
   ```

3. **Install Required Libraries**

   Create a `requirements.txt` file in the root directory with the following content:

   ```
   Flask==2.0.1
   transformers==4.16.2
   gtts==2.2.3
   fuzzywuzzy==0.18.0
   ```

   Install the dependencies using:

   ```
   pip install -r requirements.txt
   ```

4. **Run the Application**

   ```
   python chatbot_server.py
   ```

   The application will be accessible at http://127.0.0.1:5000/.

Issues and Troubleshooting

- **LF/CRLF Warning**: A warning about LF being replaced by CRLF is related to line endings in text files. This does not affect the functionality of the application.

Video Demonstration

Watch the one-minute video demonstrating the chatbot in action: Chatbot Demonstration

---

Contact

For any queries or feedback, please contact Harshavardhan at harshavardhanmanavalan@gmail.com.