Banking Chatbot Report

Overview

The provided code is a Python script implementing a banking chatbot with various functionalities such as text and speech input processing, intent recognition, account information retrieval, money transfer, and integration with a chatbot model for general conversation.

Dependencies

The code imports several libraries and packages necessary for different functionalities:

- `sounddevice`: For recording audio.
- `numpy`, `torch`, `pyttsx3`, `wave`: For various audio and speech processing tasks.
- `vosk`: A library for offline speech recognition.
- `langchain`: Custom modules for chatbot implementation.
- `os`, `json`: For file and environment handling.
- `streamlit`: For building web applications.
- `transformers`: For BERT model usage.

Functionalities

1. **Recording and Recognizing Speech:** The `record_audio` function records audio for a specified duration and saves it as a WAV file. The `recognize_speech` function uses the Vosk library to recognize speech from the recorded audio.
2. **Text-to-Speech Conversion:** The `text_to_speech` function converts text to speech using the `pyttsx3` library.
3. **Chatbot Integration:** The `ChatOpenAI` class from `langchain` module is used for generating responses to user queries. The `get_chat_model_response` function interacts with the chatbot model to get responses based on user input.
4. **Intent Recognition:** Intent recognition is done using a BERT-based model loaded with `BertForSequenceClassification`. The `llm_model_for_intent_recognition` function predicts the intent label for a given text input.

5. **Account Information Handling:** The code reads account information from a JSON file and provides functionalities to retrieve balance, dues, and perform money transfers. Functions like `extract_account_number`, `get_account_info`, `update_account_balance`, and `transfer_money` handle account-related operations.

6. **User Interaction:** The script provides a Streamlit-based user interface where users can interact with the chatbot through text input or audio recording. User inputs are processed to determine the intent and perform appropriate actions.

Usage

Users can interact with the chatbot by typing messages or recording audio. The chatbot responds based on the user's query, performs actions like fetching account information, transferring money, or engaging in general conversation.

Recommendations for Improvement

1. **Error Handling:** Implement robust error handling mechanisms, especially for user inputs and file operations.

2. **Security:** Ensure secure handling of sensitive data such as account numbers and balances.

3. **Optimization:** Optimize code for better performance, especially in audio processing and model inference.

4. **Enhanced User Experience:** Improve the user interface for better usability and clarity of actions.

Conclusion

The provided code offers a comprehensive implementation of a banking chatbot with speech and text input capabilities, intent recognition, and integration with a conversational AI model. With further refinement and enhancements, it can serve as an efficient tool for providing banking assistance and engaging with users in natural language conversations.