Project Specification:
Multi-Modal Chatbot Application

Overview

This project aims to develop a multi-modal chatbot application that enables users to engage in conversations with the chatbot using both speech and text-based inputs. The chatbot will utilize speech recognition to convert user speech into text and employ text-to-speech technology to respond to users audibly. Additionally, an advanced Natural Language Processing (NLP)-based chatbot will be integrated to enable context-aware and interactive conversations.

Problem Solving

This project aims to solve the issue of managing calls while engaged in other activities. It helps individuals avoid distractions and ensures a safer interaction with callers when answering isn't feasible.

Users

The primary users of this system are individuals who receive unwanted or important calls while occupied, such as when driving, and want an automated way to interact with the caller without physically answering. Secondary stakeholders include people calling the user, such as telemarketing companies or the user's friends or family.

Version 1: Speech-to-Text and Text-to-Speech

This initial version of the application will be designed as a web-based platform. Users can provide input in speech, which will be converted into text. Subsequently, this text will be transformed back into speech, allowing users to audibly hear the chatbot's responses. This version focuses on creating a web interface for seamless speech-to-text and text-to-speech interactions, enhancing user engagement and accessibility.

Features

- **Web-based Application:** Develop a web-based interface for user interaction.
- **Speech Recognition:** Implement speech-to-text technology to convert user speech inputs into text.
- **Text-to-Speech Conversion:** Utilize text-to-speech technology to transform chatbot responses into audible speech for users.

Purpose

- **Enhance User Accessibility:** Provide a user-friendly platform for speech-based interactions.
- **Improve User Experience:** Ensure clear and comprehensible communication by converting text responses into speech.

Task Vignettes (User activity "flow")

Task 1: User Input

Description: The user initiates a conversation by sending a message to the app.
Steps:

1. User opens the chat interface within the app.
2. User speaks or types a message to communicate with the chatbot.
3. User clicks on a "Send" button to submit the message.

Task 2: Speech-to-Text Conversion

Description: The app employs speech-to-text technology to convert the spoken message into text for processing.
Steps:

1. The app receives the user's spoken message.
2. The app utilizes speech recognition to convert the spoken message into text.
3. The converted text message is displayed on the screen.

Task 3: Text Response

Description: The app displays the converted text message, allowing the user to respond via text.
Steps:

1. The user views the converted text message displayed on the app's interface.
2. The user formulates a response by typing a text message.
3. The user clicks on a "Send" button to submit the text response.

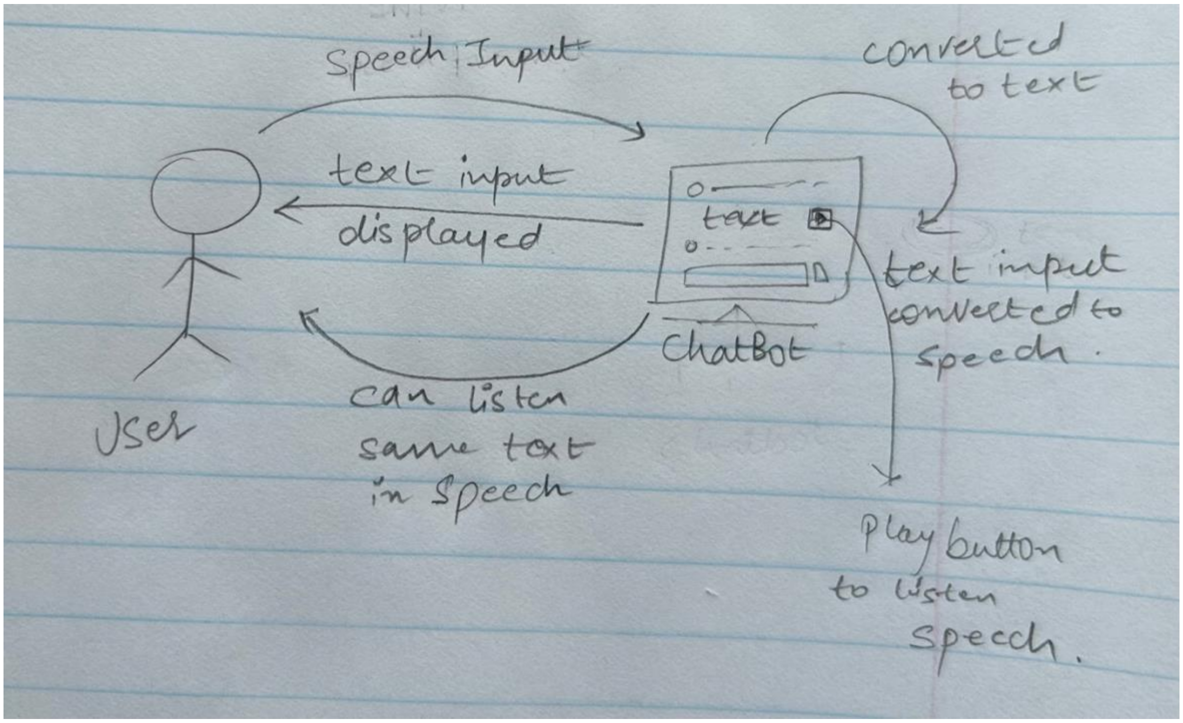
Task 4: Text-to-Speech Conversion

Description: The user-generated text response is transformed back into speech using text-to-speech technology.
Steps:

1. The app receives the user's text response.
2. The app utilizes text-to-speech technology to convert the text response into audible speech.
3. The synthesized speech response is prepared for playback.

Technical Flow

1. Develop a web-based interface using HTML, CSS, and JavaScript to interact with users.
2. Implement a speech recognition module (e.g., using libraries or APIs) to capture user speech inputs.
3. Utilize speech recognition technology to convert captured speech into text format.
4. Process the text input and forward it to the chatbot module for further handling.
5. Utilize text-to-speech technology to transform chatbot responses into audible speech.
6. Users can communicate with the chatbot using both speech and text inputs.
7. The chatbot provides audible responses to users' queries and statements.


Speech Input
converted to text
text input displayed
text
ChatBot
text input converted to speech.
User
can listen same text in Speech
Play button to listen speech.

Technologies and Tools

- Speech Recognition: Utilize speech recognition libraries or APIs (such as Whisper).
- Text-to-Speech: Integrate text-to-speech engines for speech synthesis.
- NLP Frameworks: Implement NLP using Python libraries (e.g., PocketSphinx, TTS engine, etc.).
- Web Development: Create a user-friendly web interface using HTML, CSS, and JavaScript.
- Version Control: Use Git for project management and version control.

Version 2: NLP-Based Chatbot

In the second iteration of this application, an NLP-based chatbot will be implemented. This chatbot will facilitate dynamic two-way conversations between users and the system. Users can continue to utilize speech and text recognition for input, but now they will also experience advanced natural language processing capabilities, enabling context-aware and context-rich interactions. This version elevates user engagement and delivers more intuitive and meaningful conversations.

Learning Goals

- Gain Proficiency in Speech-to-Text and Text-to-Speech Technologies.
- Learning Natural Language Processing (NLP) and Chatbot Development.
- Understand the Challenges and Opportunities in Multi-Modal Conversational Interfaces.

Self-Assessment

After working through the spec, what was the biggest, most unexpected change to had to make from your sketch?

The development of this project for me is learning a small part of the bigger picture. But there are so many open ends and confusions that will take time to be filled. Therefore, the biggest challenge was to decide on which part I should focus and develop. Also, I added an NLP feature for version 2, which I hope I can complete in this time frame.

How confident do you feel that you can implement the spec as it's written right now?

I have written the spec file by searching various resources on how to build this application. So, I would say that I am not 100% confident that the application will be completely according to the specs. I might have to modify certain technologies or plugins for a feasible solution.

What is the biggest potential problem that you NEED to solve (or you'll fail)?

I think building an NLP chatbot will be a challenge. Currently, I am not sure about the training of the bot, or maybe I can research certain APIs, such as the one available by open for developers.

What parts are you least familiar with and might need my help?

I am not familiar with any plugins or modules that I can utilize to build this application. Therefore, any suggestions might be very helpful.