

SFBU Customer Support System - Speech to Text to Speech

Special Topics: Generative AI-Driven Intelligent Apps
Development

Ademilton Marcelo da Cruz Nunes (19679)

Links

Github:

<https://github.com/ademiltonnunes/Machine-Learning/tree/main/ChatGPT/Custom%20Support%20System/SFBU%20Customer%20Support%20System%20-%20Speech%20to%20Text%20to%20Speech>

Table of Content

- Introduction
- Introduction - Integration
- Implementation
- Implementation - Asking questions with voice
- Recording Audio
- Transcribing Audio to Text
- Answering question with ChatGPT
- Transcribing Text to Audio
- Conclusion

Introduction

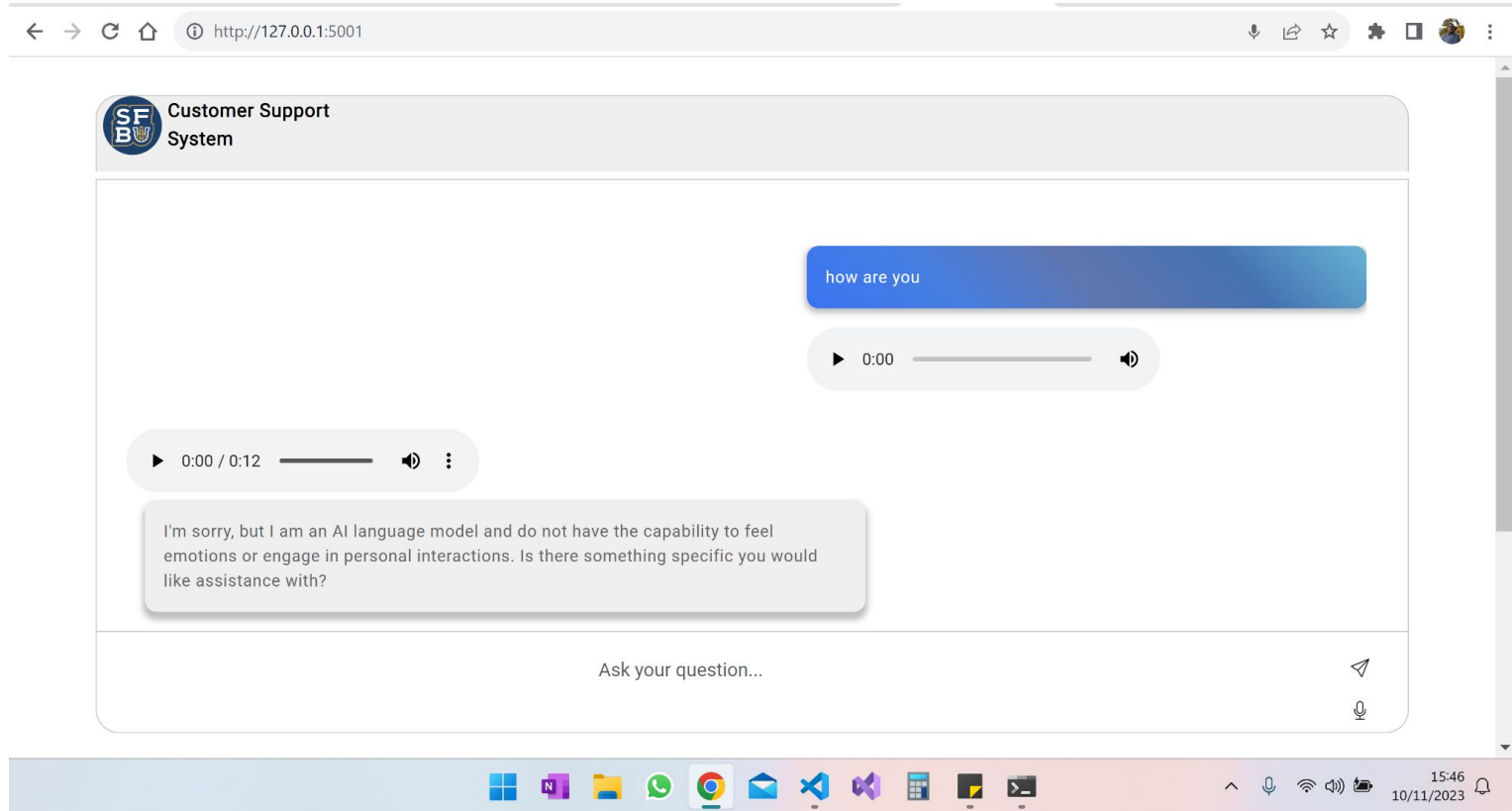
This project aims to integrate two projects:

- The first project: it is a web application of a customer support system for SFBU that answers customer's questions based in a loading PDF. This project can be found in:
<https://github.com/ademiltonnunes/Machine-Learning/tree/main/ChatGPT/Customer%20Support%20System/SFBU%20Customer%20Support%20System%20-%20Chatbot%20From%20Files>
- The second project: it is implement a AI-Based “Alexa”, which is the possibility of speaking with ChatGPT with voice using Whisper and Google Text to Speech (GTTS). This project can be found in:
<https://github.com/ademiltonnunes/Machine-Learning/tree/main/ChatGPT/Real-time%20Speech%20to%20Text%20to%20Speech>

Introduction - Integration

This project implemented the possibility of asking questions in a chatbot also through audio. Also, the system also answers questions in voice and text.

Implementation



Implementation - Asking questions with voice

By pressing the microphone icon, the user clicks to ask questions in audio.

When the user finishes asking, he/she clicks on "stop recording".

When the recording ends, it is automatically submitted to the chatbot.

This voice question is answered in the same way that text questions are asked

Implementation - Answering questions with voice

When a voice message is received, the system responds to the question with text and voice.

This response is shown on the screen with the option to play the system's audio response by clicking the play button.

Recording Audio

Audio recording is done on the client side (browser). Audio recording is done by an asynchronous JavaScript function.

First, I check if the customer has a microphone and if it is authorized to be used. The recording is made with MediaRecorder instance, which is an API for recording media streams. This recorder is initialized with the microphone audio stream obtained from the user.

The recorded audio data is in the form of "data chunks". When the recording is stopped (either by the user), this event handler is triggered. It creates a Blob object from the accumulated audio chunks, specifying the MIME type as 'audio/mp3'.

Recording Audio

```
async function startRecording() {
  const startRecordingButton = document.getElementById('startRecordingButton');
  const stopRecordingButton = document.getElementById('stopRecordingButton');

  let audioChunks = [];

  // Getting user permission to access the microphone
  navigator.mediaDevices.getUserMedia({ audio: true })
    .then((stream) => {
      mediaRecorder = new MediaRecorder(stream);

      //Change buttons
      startRecordingButton.style.display = 'none';
      stopRecordingButton.style.display = 'inline-block';
      console.log('Recording started...');

      mediaRecorder.start();

      mediaRecorder.ondataavailable = (event) => {
        if (event.data.size > 0) {
          audioChunks.push(event.data);
        }
      };

      mediaRecorder.onstop = () => {
        const audioBlob = new Blob(audioChunks, { type: 'audio/mp3' });
        // Send `audioBlob` to server
        sendAudioToServer(audioBlob);
        // Clear mediaRecorder
        mediaRecorder = null;
      };
    })
    .catch((error) => {
      alert('Error accessing microphone:', error);
      console.error('Error accessing microphone:', error);
    });
}
```

Transcribing Audio to Text

The audio recorded by the client is sent to the server. The audio is stored in a local directory on the server.

Using Whisper, the base module, the audio is transcribed into text. Punctuation and capital letters are also added to the text when necessary.

Transcribing Audio to Text

```
def transcribe_audio(self, audio_path, model = "base") -> str | Any |  
try:  
    # Load model  
    audio_model = whisper.load_model(model)  
  
    # Transcribe audio  
    result = audio_model.transcribe(audio_path, language='english')  
  
    #Extract text  
    predicted_text = result["text"]  
  
    #Punctuation  
    punc = ' '!()-[]{};: '\", <> . / ? @ # $ % ^ & * _ ~ '''  
    predicted_text = predicted_text.translate({ord(i): None for i  
    # Capitalize the first letter  
    predicted_text = predicted_text.capitalize()  
  
    return predicted_text  
except Exception as e:  
    e = ("Error transcribing audio:", str(e))  
    return e
```

0 Live Share

Ln 123, Col 52

Spaces: 4

UTF-8

CRLF

{ Python

3.11.5 64-bit



16:11
10/11/2023



Answering question with ChatGPT

With the audio transcribed into text, the text is sent to LLM to answer the question based on the uploaded pdf.

```
def chat(self, question) -> Dict[str, Any] | Exception:
    try:
        chatbot = self.__load_chatbot()
        response = chatbot({'question': question, 'chat_history': []})
        return response
    except Exception as e:
        return e
```

Ln 127, Col 9 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit

16:14 10/11/2023

Transcribing Text to Audio

With the answer given by the LLM, the answer is submitted to Google Text to Speech (GTTS). The response is transformed into audio and sent to the client.

The text and audio response is sent to the client (user browser).

Transcribing Text to Audio

```
def text_to_audio(self, text) -> gTTS:  
    mp3_obj = gTTS(text=text, lang="en", slow=False)  
    return mp3_obj
```

0  Live Share Ln 147, Col 1 Spaces: 4 UTF-8 CRLF {Python} 3.11.5 64-bit 



16:20
10/11/2023



Conclusion

This project demonstrated the integration between two projects, first project SFBU Customer Support System - Chatbot From Files, and second the project implements an AI-Based “Alexa”.

I applied techniques of recording audio, transcribing audio to text and transcribing text to audio. I showed solid examples of how to implement those techniques successfully.