Voicebot for Visually Impaired Students

Undergraduate Report

Submitted in partial fulfillment of the requirements of
CS F367 Laboratory Project

By

Kavyanjali Agnihotri
ID No. 2020A7PS0185H

Under the supervision of:

Dr. Dipanjan Chakraborty

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, HYDERABAD CAMPUS
December 2023

Declaration of Authorship

I, Kavyanjali Agnihotri, declare that this Undergraduate Report titled, 'Voicebot for Visually Impaired Students' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Certificate

This is to certify that the thesis entitled, "*Voicebot for Visually Impaired Students*" and submitted by Kavyanjali Agnihotri ID No. 2020A7PS0185H in partial fulfillment of the requirements of CS F367 Laboratory Project embodies the work done by him under my supervision.

Supervisor
Dr. Dipanjan Chakraborty
Assistant Professor,
BITS-Pilani Hyderabad Campus
Date:

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, HYDERABAD CAMPUS

Abstract

Bachelor of Engineering (Hons.) in Computer Science & Information Systems

Voicebot for Visually Impaired Students

by Kavyanjali Agnihotri

This report explores the use of voice chatbots to enhance the education and communication experiences of visually impaired children. It primarily focuses on using technologies like Amazon Alexa and Microsoft Azure AI capabilities to build specialised voice chatbots. The report analyses the potential of voice chatbots to alleviate the difficulties faced by visually impaired students in traditional schooling. In addition to reviewing the benefits and drawbacks of using Amazon Alexa, the report highlights the advantages of Microsoft Azure AI services, particularly Language Studio and Speech Services. It also covers ideas for further research and testing in the real world with the aim of developing a welcoming and open educational resource for kids who are blind or visually impaired. Furthermore, we try to use a Raspberry Pi as an interface and make the code compatible for the same.

Acknowledgements

First of all, I want to express my gratitude to Birla Institute of Technology and Science, Pilani for providing me with the chance to work on the CF367 Laboratory Project's "Chatbots for Blind Children" theme. I want to sincerely thank Dr. Dipanjan Chakraborty, Assistant Professor at the BITS Pilani Hyderabad Campus, for providing me with this opportunity and serving as my project adviser. His genuineness, enthusiasm, and commitment have greatly motivated me. I learned the methodology to complete the task and deliver the work as efficiently as possible from Dr. Dipanjan Chakraborty. I express him a huge degree of gratitude for educating me during every encounter. Being able to work and learn under his direction is a huge honour and privilege.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Technology has become an effective tool in today's society for overcoming barriers and giving people with different capacities equal chances. The creation of voice chat bots that are especially suited to meet the unique needs of blind children, with a focus on assisting their learning and communication, is one exciting area of innovation.

1.2 Motivation

The journey of a visually impaired child is marked by challenges that extend far beyond the classroom, impacting their daily lives and social interactions. Conventional educational approaches often fall short in addressing the distinctive requirements of blind children, leaving them at a disadvantage.

1.3 The Promise of Voice Chat Bots

Voice chat bots can be meticulously designed for visually impaired individuals[21][19]. They come equipped with remarkable speech recognition and synthesis capabilities. This enables them to engage with the child through spoken language, effectively serving as versatile educational aides. By relying on auditory cues and voice-based interactions[9], these voicebots[10] can make the process of learning interactive and engaging, as well as nurturing independence and self-confidence in blind children.

1.4 Project Objectives

The goal of this project is to explore the creation and use of voice chat bots that are specifically designed to meet the needs of blind children, especially in the area of education and that can be further developed to provide tests and other interactive activities. This project also tries to concentrates on the most practical and affordable chatbot device.

Chapter 2

Current Solutions and Technologies

2.1 Introduction

This chapter can discuss the existing strategies and technologies in place to address the challenges faced by visually impaired students along with a sort of literature review.

2.2 Challenges in Schools

2.2.1 Understaffing

In the education of visually impaired students, there are multiple challenges faced by the students as well as the teachers. One striking issue is the severe understaffing of schools, which directly impacts the quality of education and support provided to these students. This scarcity of teachers exacerbates the difficulties faced by visually impaired learners, as they need specialized materials and supervision.

2.2.2 Inaccessibility of Touch Sensors

Additionally, touch sensors—a vital tool for children who are blind or visually impaired—are expensive and in short supply. With the help of these sensors, students can access information through tactile feedback, which helps their comprehension and engagement with the course materials. Unfortunately, the lack of such equipment makes it difficult for visually impaired pupils to learn and puts them at a disadvantage.

2.2.3 Lack of Braille textbooks

The lack of braille textbooks is a serious issue as well. These texts are not only hard to get, but they are also expensive. The equal access to educational resources is impeded by this financial barrier. Even while computers might be able to provide solutions, their availability and accessibility remain problematic.

2.3 Realising Potential Solution - Voice Chatbots

Recognising that voice is global and creating voicebots or voice-enabled technology[9][10][21][19] will greatly improve the problem. Through the use of voice-based interfaces and tools, we hope to close the educational gap for students who are blind or visually challenged. Additionally, it benefits pupils who don't necessarily have visual problems.

2.4 Organization in this Area

Microsoft's SEEDS[18] and the non-profit group Vision Empower Trust[24] are two organisations at the forefront of this trend.

2.4.1 SEEDS

SEEDS stands for Scalable Early Education with Digital Scaffolding[18]. The usage of an Interactive Voice Response (IVR) system is a crucial component of this. With the help of this system, teachers and students can connect to numerous classrooms at once, fostering an engaging and diverse learning environment. Additionally, it gives teachers the ability to keep track of student attendance and guarantee that visually impaired pupils receive the assistance they need to succeed in their academics.

2.4.2 Vision Empower Trust

This is non-profit group[24] which was initially founded in Karnataka. They are committed to empowering those who are blind or visually challenged. It actively works to make education accessible to everyone and offers instructional content in Kannada.

Chapter 3

Methodology

3.1 Introduction

Our process for developing a voice chat bot customised to the special requirements of blind children involves a thorough analysis of various platforms and technologies. This procedure was designed to find the best options for our project. We go into the procedures used to obtain data and carry out research below. We explore Amazon Alexa, Microsoft Azure AI related services, Google Assistant, Rasa, Amazon Lex and Poly. Along with some platforms which were discovered to be deprecated, these are Google Nest and Microsoft LUIS and CLU services.

Further we tried to make a voicebot using RaspberryPi and required electronics, this also was filled with challenges in terms of availability of the items as well as code incompatibility.

3.2 Voicebot Platforms

3.2.1 Amazon Alexa

Amazon Alexa[4] is a cloud-based virtual assistant developed by Amazon. It is designed to interact with users through voice commands and provide a wide range of services and information. Alexa is often used through Amazon Echo smart speakers and other compatible devices.

3.2.1.1 Exploration

Our exploration of voice chatbot platforms began with the development of a skill on the Amazon Alexa platform. This skill was designed to recognize the entity value spoken by the user, and find the corresponding uploaded cloud URL value on MongoDB. This URL needs to be publicly

accessible and should allow other services (here, Alexa) to play audio content. The skill finally aimed to play the audio file corresponding to the named entity.

3.2.1.2 Development Challenges

Our work started with the playing of text uploaded corresponding to entities in MongoDB. This involved manual addition of slot values i.e. values recognized by Alexa into the skill, without which the Alexa won't be able to reply appropriately. Furthermore, various sample utterances of the user had to be added corresponding to the intent which made it limiting to recognize different utterances unless it was on the system. Overall, we were able to extract the slot value from the spoken sentence, get the value from mongodb for that slot value and play it as a reply.

The figure 3.1 shows the invocation of the developed skill.

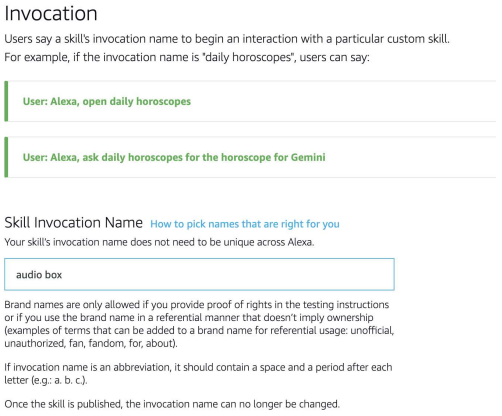

FIGURE 3.1: Skill Invocation

The figure 3.2 shows the intent handler code written to handle topic slot values

```
const TopicIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
            && Alexa.getIntentName(handlerInput.requestEnvelope) === 'TopicIntent';
    },
    async handle(handlerInput) {
        const speakOutput = 'Hello! What topic do you want to know about?';
        const wordSlotValue = Alexa.getSlotValue(handlerInput.requestEnvelope, 'topic');
        var speechOutput = ''

        if(wordSlotValue === undefined) {
            speechOutput = "Sorry, I do not have information about it.";
            return handlerInput.responseBuilder
                .speak(speechOutput)
                .reprompt(speakOutput)
                .getResponse();
        }

        const client = new MongoClient(MONGODB_URL, { useNewUrlParser: true, useUnifiedTopology: true });
        await client.connect();
        const db = client.db(DATABASE_NAME);
        const collection = db.collection(COLLECTION_NAME);

        const result = await collection.findOne({ name: wordSlotValue });
        if (result) {
            const description = result.desc;
            speechOutput = `The description for ${wordSlotValue} is: ${description}. What other topic do you want to know about?`;
        } else {
            speechOutput = `Sorry, I couldn't find a description for ${wordSlotValue}. What other topic do you want to know about?`;
        }
        client.close();

        return handlerInput.responseBuilder
            .speak(speechOutput)
            .reprompt(speakOutput)
            .getResponse();
    }
```

FIGURE 3.2: Code for Topic Intent

3.2.1.3 Cloud File Upload Providers

In order to play audio files, we explored a number of file upload providers namely Microsoft Azure, Amazon S3 bucket, Google Cloud, Google Drive, Microsoft OneDrive, and many more. The links of the audio files need to be publicly accessible in order for alexa to play it. Majority of these were paid service providers or needed a paid plan to make a link publicly accessible as needed by Alexa. For this, we were able to use just Microsoft Azure.

3.2.1.4 MongoDB Integration

Adding onto the work already done, the values in MongoDB were replaced by the publicly accessible URL for the recognized slot value or entity, which was played by Alexa as desired.

3.2.1.5 Advantages

Amazon Alexa offered several advantages, making it an attractive choice for our initial exploration. Notably, it proved to be cost-effective, entailing low setup costs and straightforward skill development processes.

3.2.1.6 Drawbacks - Language and Complexity

To make the skill compatible with various Indian languages, support for each language has to be added, and multiple files need to be modified and filled by the other language equivalents. This process seems very time-consuming and redundant. Moreover, the process required manual addition of slot values (see 3.3, i.e., the values recognized by Alexa, in our case entities. This made it difficult to recognize new or different words if they were not present in the slot values added during the development period. Therefore, these values were only recognized if the user spoke in a particular manner for which we had to manually add the sample utterances (see 3.4). All of this makes Alexa unnecessarily complex and burdensome to use.

Another significant drawback was its limited language support, restricted to English (see 3.5) and Hindi.

Additionally, at random times, Alexa spoke words that were not uploaded by us.

Finally, due to privacy issues, Alexa does not provide us with the complete sentence spoken by the user, which makes it difficult to understand the intent of the user.

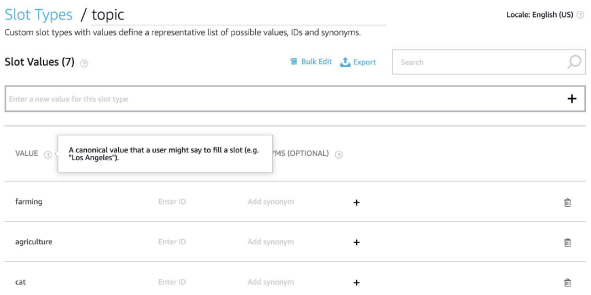

FIGURE 3.3: Slot values inserted manually

FIGURE 3.4: Sample Utterances for Alexa

3.2.2 Microsoft Azure AI services

In our project, we leveraged Microsoft Azure AI Services[11], a suite of powerful tools and capabilities offered by Microsoft, to explore the development of a voice chat bot tailored for blind children. The Azure AI Services played a vital role in various aspects of our project, including Language Studio , Speech Services, and in future hopefully Keyword (or Wake word) Recognition.

3.2.2.1 Language Studio: Conversational Language Understanding

The Microsoft Azure Language Studio[25] is a versatile tool that enables developers to create custom language models for various natural language understanding tasks. In our project, we harnessed the capabilities of Language Studio to train a conversational language understanding (CLU) model. This model was specifically designed to recognize slot values spoken by users within sentences. The model was trained on examples of Hindi and English language. We require a decent amount of sample utterance for the model to perfectly predict the desired slot value and

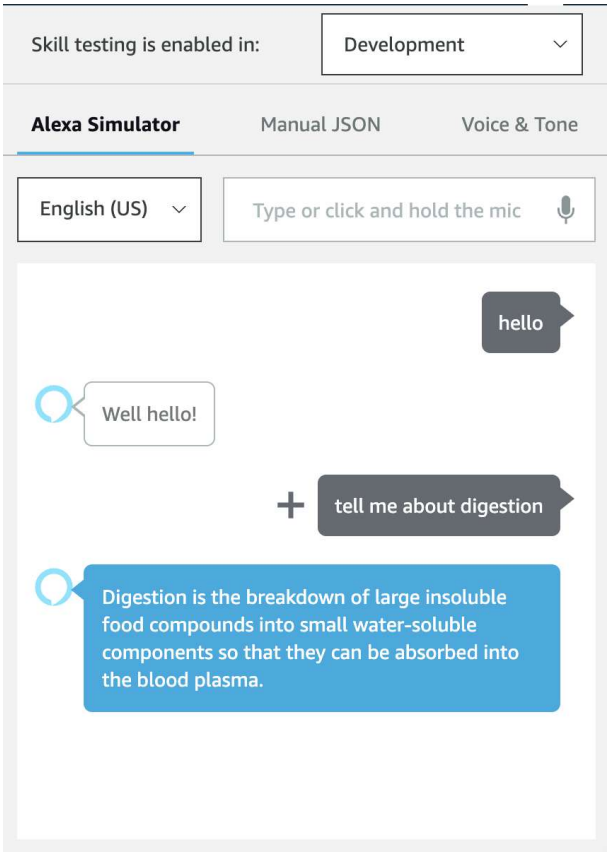

FIGURE 3.5: Alexa Test Interface

the slot value with the highest probabilty is the slot value as shown in 3.6 and 3.7. Moreover, we need to provide sample utterances in all the languages we want the model to be used on. Some sample utterances are given in 3.8 and 3.9

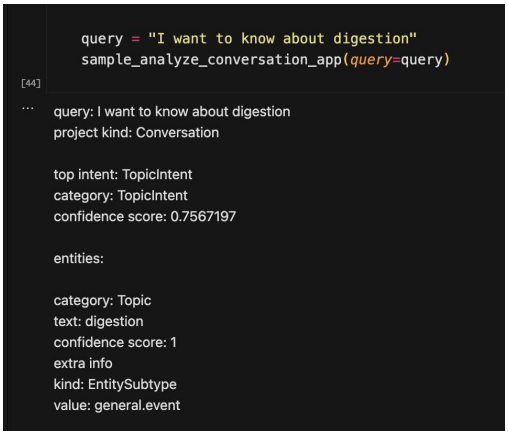

FIGURE 3.6: Slot Value Predication for English Utterances

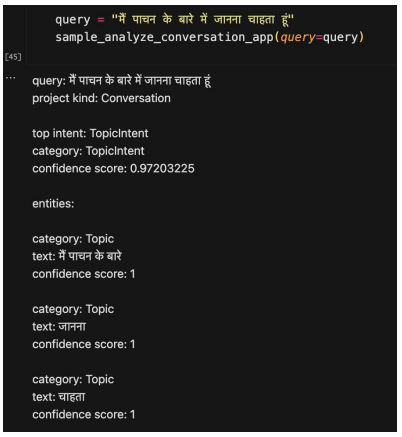

FIGURE 3.7: Slot Value Predication for Hindi Utterances

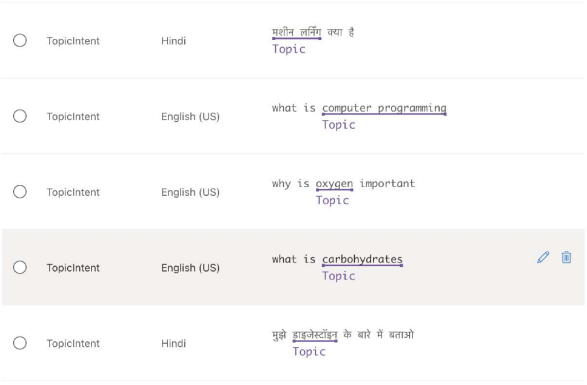

FIGURE 3.8: Sample Utterances for CLU

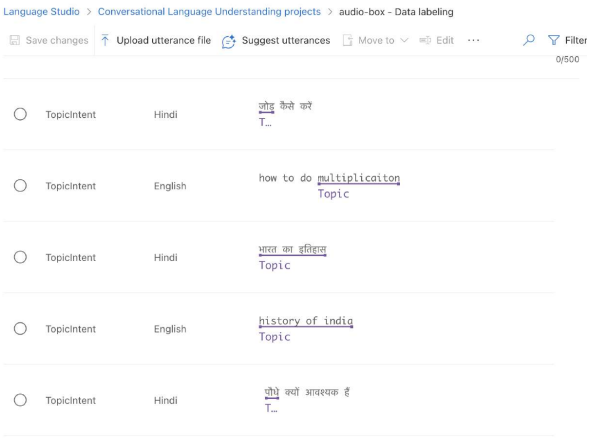

FIGURE 3.9: Sample Utterances for CLU #2

3.2.2.2 Speech Services: Enhancing Voice Interaction

This provided us with powerful speech-to-text[20] (see 3.10 and for usage see 3.11) and text-to-speech[23] (see 3.12). These services allowed us to convert spoken language into written text and vice versa, enabling seamless interaction between the chat bot and the users. This is crucial in making the chatbot accessible and user-friendly for visually impaired children, as it facilitated voice-based communication. For language support refer to A.

3.2.2.3 Translation for Multilingual Support

We can also translate from some languages (see 3.13 and 3.14) to another as per the documentation given. Furthermore, we can use the translation tool to convert into English so that fetching MongoDB data is easier as well as the data uploaded is not redundant.

Using all of these services together, we can get the following backend flow as per 3.15.

```python
def recognize_from_microphone(speech_language):
    speech_config = speechsdk.SpeechConfig(subscription=SPEECH_KEY, region=SPEECH_REGION)
    speech_config.speech_recognition_language=speech_language

    audio_config = speechsdk.audio.AudioConfig(use_default_microphone=True)
    speech_recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config, audio_config=audio_config)

    print("Speak into your microphone.")
    speech_recognition_result = speech_recognizer.recognize_once_async().get()

    if speech_recognition_result.reason == speechsdk.ResultReason.RecognizedSpeech:
        print("Recognized: {}".format(speech_recognition_result.text))
    elif speech_recognition_result.reason == speechsdk.ResultReason.NoMatch:
        print("No speech could be recognized: {}".format(speech_recognition_result.no_match_details))
    elif speech_recognition_result.reason == speechsdk.ResultReason.Canceled:
        cancellation_details = speech_recognition_result.cancellation_details
        print("Speech Recognition canceled: {}".format(cancellation_details.reason))
        if cancellation_details.reason == speechsdk.CancellationReason.Error:
            print("Error details: {}".format(cancellation_details.error_details))
            print("Did you set the speech resource key and region values?")
```

FIGURE 3.10: Speech-to-Text Code

```
recognize_from_microphone(ENGLISH)

Speak into your microphone.
Recognized: Hi, what are you doing?

recognize_from_microphone(HINDI)

Speak into your microphone.
Recognized: हैलो, आप क्या कह रहे हो?
```

FIGURE 3.11: Speech-to-Text Example

```
text = "Tell me about ducks"
voice = "en-US-AriaNeural"
synthesize_text(text, voice)
[27]
...  Speech synthesized for text [Tell me about ducks]

text = "मुझे बत्तखों के बारे में बताएं"
voice = "hi-IN-SwaraNeural"
synthesize_text(text, voice)
[28]
...  Speech synthesized for text [मुझे बत्तखों के बारे में बताएं]
```

FIGURE 3.12: Text-to-Speech Examples

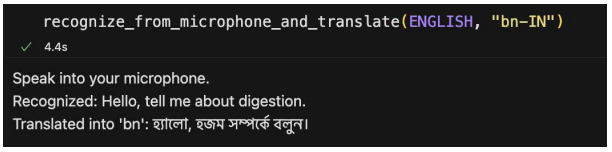

FIGURE 3.13: Translation Service (English to Bengali)

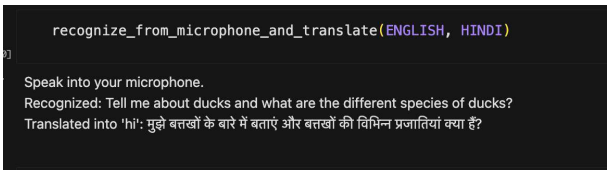

FIGURE 3.14: Translation Service (English to Hindi)

3.2.2.4 Exploring Keyword Model

We are going to experiment with keyword or wake word[12] recognition. This involved training models to identify specific keywords or phrases spoken by users. This feature will be especially beneficial in improving the bot's responsiveness to specific prompts and usability.

3.2.2.5 Advantages

Support for Many Indian Languages: A significant advantage of using Microsoft Azure AI Services was its extensive language support. In addition to Hindi, Azure offers support for several other Indian languages, these are given in A.

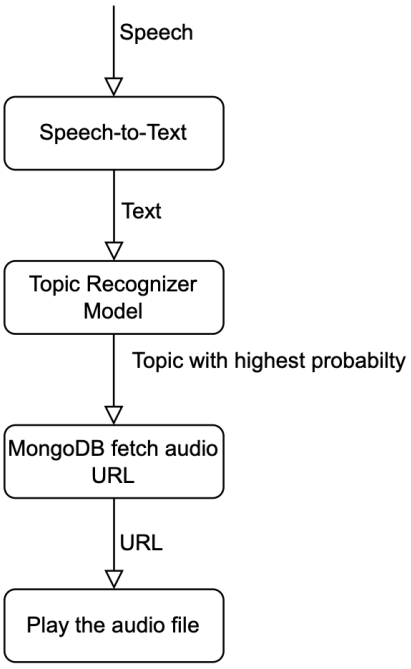

FIGURE 3.15: Predicated Server Side flow

3.2.2.6 Challenges

Dubious Interface: One notable challenge encountered during the project was the somewhat unclear way to make a frontend for this service. This ambiguity sometimes made it more time-consuming to navigate through the various services and functionalities. This is especially the case in web-based interfaces.

Valid and Sufficient Data for Topic Recognizing: In order to get correct topic predictions for the 11 supported languages, there needs to be sufficient data[22] for the model to be trained on. Acquiring this data is quite difficult and one alternative of this is to use translation service in the very beginning as depicted in the 3.16

Limitation for Translation Services: The free tier version of Azure Translation services only provides us with 2 Million words per month. This is a huge limitation as we aim to use these voicebots in numerous quantities.

Bot Framework: Microsoft Azure offers the Bot Framework (see 3.17), which is a powerful tool for building and deploying chatbots. However, exploring this framework posed its own set of challenges. In order to make it voice-enabled, we need to have a Windows device for the voice

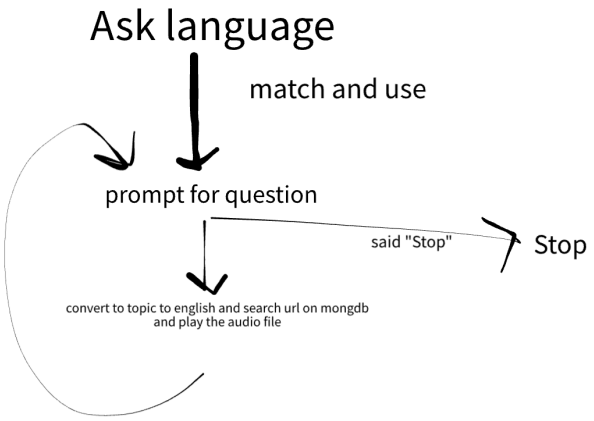

FIGURE 3.16: Revised Flow as an Alternative

assistant testing service. Nonetheless, the potential benefits in terms of chatbot capabilities and integration with a wide range of platforms made it a promising avenue.

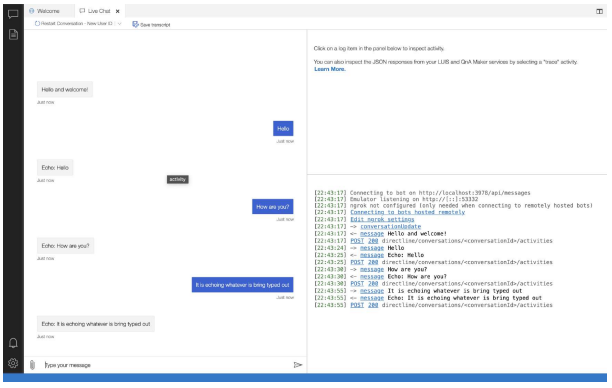

FIGURE 3.17: Simple EchoBot using Bot Framework

3.2.2.7 Summary

In summary, Microsoft Azure AI Services played a pivotal role in our project by providing a robust foundation for language understanding, speech processing, and keyword recognition. The platform's extensive language support, albeit with an occasionally challenging interface, opened doors for creating a more inclusive and versatile voice chat bot for blind children. Furthermore, exploring alternatives to free translation service is imperative.

3.3 Other Platforms

3.3.1 Google Assistant

We explored the Google Assistant platform to understand its compatibility with Bluetooth devices and determined whether additional hardware, such as smartphones or tablets, would be necessary. Advantages included support for multiple languages and compatibility with Android devices. On the downside, developing an app action on Google Assistant required app publication on Google Store, developer fees amounting to $25, and potential limitations in device compatibility.

3.3.2 Rasa

Rasa[13] had challenges which included unclear language support, the potential need for speech-to-text and text-to-speech interfaces, and a web-based interaction model.

3.3.3 Amazon Lex and Polly

We explored Amazon Lex[5] and Amazon Polly[6] as a potential platform for chatbot development. Challenges included uncertain language support, potential need for credits, and limited information on device support.

3.3.4 Deprecated Platforms

Google Nest: Google Nest is deprecated and no longer a viable option for our project.

Microsoft LUIS: Likewise, Microsoft LUIS was found to be deprecated and was not considered in our analysis.

3.4 Tranlation Services: Alternatives

3.4.1 Bhashini

Among one of the free translation services available is Bhashini[8]. Unfortunately, there is no clear indication on how to use its APIS. Therefore, this seemed to be a dead end.

3.4.2 AI4Bharat

Another one of the Open Source serivces AI4Bharat. It has two models for translation task i.e. IndicTrans[3] and IndicTrans2[2]. Both of these models are available on HuggingFace.

In order to use these models, we need an API through which requests can be sent as the model itself can't be deployed on the RaspberryPi[16] as it requires a lot of compute and resources. After getting in touch which the authors, they provided an alternative to deploy a distilled version of the model on CPU and somehow use it.

3.5 Internet of Things: Raspberry Pi

The development of the frontend interface for the voicebot involved a combination of Raspberry Pi hardware and various electronic components along with software implementation using Python. However, due to compatibility issues with certain libraries on the Raspberry Pi, an exploration of an alternative solution utilizing the .NET framework[1] was undertaken.

3.5.1 Hardware Configuration

The hardware setup primarily comprised a Raspberry Pi 4b board running Raspbian OS, equipped with essential peripherals including a microphone, speakers, and an SD card flashed with Raspbian. The Raspberry Pi was connected to the interface using an HDMI cable to facilitate display connectivity. Additionally, the Raspberry Pi was configured with a static IP address for streamlined network communication and accessibility. Procuring essential hardware components posed a significant challenge throughout the development process. Acquiring specific electronic elements especially microphones became arduous due to prevalent instances of faulty or incompatible products in the market. Multiple iterations were necessary to secure functional hardware that met project requirements. Dealing with discrepancies in specifications, encountering faulty items, and addressing compatibility issues caused substantial delays, disrupting the scheduled timeline for the implementation phase.

3.5.2 Python Implementation

Initially, the interface development was initiated using Python due to its ease of integration with Raspberry Pi[14][15][17] and the availability of diverse libraries for voice recognition and interaction. The Python codebase facilitated the integration of the microphone, speakers, and other components, enabling voice-based interactions and responses within the frontend interface.

3.5.3 Challenges and Transition to C#

However, challenges arose as certain essential libraries i.e. 'azure-cognitiveservices-speech'[7] crucial for the functionality of the voicebot were not available and lacked compatibility with the Raspberry Pi environment. As a result, the need to explore an alternative solution utilizing the .NET framework arouse. The transition involved a migration of the existing codebase and logic from Python to .NET, while ensuring compatibility with the hardware components and maintaining the intended functionality of the interface.

All of the code can be found on this git repository.

Chapter 4

Future Work

4.1 Conclusion

Through this comprehensive and detailed exploration of various voice chat bot platforms and technologies, our intention was to acquire valuable insights into their strengths, weaknesses, and alignment with our mission of creating an inclusive and effective learning tool for blind children. In order to make it economically, we have first tried using a raspberry pie and got inspired from the ideas of the GitHub repository . The current backend is compatible a number of frontend but we are unable to use it with Raspberry Pi. In order to make a working backend compatible with Raspberry Pi turned out to be a difficult task. This research serves as the cornerstone for our subsequent phases of development and deployment.

4.2 Future Work

We plan to completely convert the backend to .NET framework so that it can be used with Raspberry Pi as shown for Closed Captioning[1].

Further, we aim to develop a prototype and do a test run in an actual setting by collaborating with the wonderful team at Vision Empower.

Appendix A

Language Support for Speech-to-Text

These are the Indian Languages supported by Microsoft Azure AI's Speech-to-Text service:

Language Code	Language	Type
`am-ET`	Amharic (Ethiopia)	Plain text
`bn-IN`	Bengali (India)	Plain text
`gu-IN`	Gujarati (India)	Plain text
`hi-IN`	Hindi (India)	see below
`kn-IN`	Kannada (India)	Plain text
`ml-IN`	Malayalam (India)	Plain text
`mr-IN`	Marathi (India)	Plain text
`pa-IN`	Punjabi (India)	Audio + human-labeled transcript
`ta-IN`	Tamil (India)	Plain text
`te-IN`	Telugu (India)	Plain text
`en-IN`	English (India)	see below

Hindi (India): Audio + human-labeled transcript, Plain text, Phrase list
English (India): Audio + human-labeled transcript, Plain text, Structured text, Pronunciation, Phrase list

Bibliography

[1] *A Closed Captioning Example on RaspberryPi using Azure Cognitive services.* URL: https://github.com/m-mohsin-ali/closed-captioning-azure-speech-ai.

[2] et la. AI4Bharat Jay Gala. *IndicTrans2: Towards High-Quality and Accessible Machine Translation Models for all 22 Scheduled Indian Languages.* URL: https://arxiv.org/abs/2305.16307.

[3] et la. AI4Bharat Jay Gala. *Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages.* URL: https://arxiv.org/abs/2104.05596.

[4] *Amazon Alexa.* URL: https://developer.amazon.com/en-GB/alexa.

[5] *Amazon Lex.* URL: https://aws.amazon.com/lex/.

[6] *Amazon Poly.* URL: https://aws.amazon.com/polly/.

[7] *azure-cognitiveservices-speech library.* URL: https://pypi.org/project/azure-cognitiveservices-speech/.

[8] *Bhashini.* URL: https://bhashini.gov.in/.

[9] et la. M Kathirvelu A Janaranjani. *Voice Recognition Chat bot for Consumer Product Applications.* URL: https://ieeexplore.ieee.org/document/9915884.

[10] et la. Meet Popat Aayush Doshi. *Alexis : A Voicebased Chatbot using Natural Language Processing.* URL: https://www.ijert.org/research/alexis-a-voicebased-chatbot-using-natural-language-processing-IJERTV11IS040155.pdf.

[11] Microsoft. *Azure AI Services.* URL: https://azure.microsoft.com/en-us/products/ai-services?activetab=pivot:azureopenaiservicetab.

[12] *Quickstart: Create a custom keyword.* URL: https://learn.microsoft.com/en-us/azure/ai-services/speech-service/custom-keyword-basics?pivots=programming-language-csharp.

[13] *RASA:Build ML-powered Conversational AI.* URL: https://rasa.community/.

[14] *Raspberry Pi 4 - How to Setup Get Started.* URL: https://www.youtube.com/watch?v=2RHuDKq7ONQ.

[15] *Raspberry Pi 4 Getting Started.* URL: https://www.youtube.com/watch?v=BpJCAafw2qE.

[16] *Raspberry Pi 4b.* URL: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/.

[17] *Raspberry Pi AI: Picroft Voice Assistant.* URL: https://www.youtube.com/watch?v=eg_V56BUmh0.

[18] *Scalable Early Education with Digital Scaffolding.* URL: https://www.microsoft.com/en-us/research/project/scalable-early-education-with-digital-scaffolding-seeds/#:~:text=SEEDS%2C%20Scalable%20Early%20Education%20with,the%20digital%20world%20to%20them..

[19] et la. Shubham Melvin Felix Sumer Kumar. *A Smart Personal AI Assistant for Visually Impaired People.* URL: https://ieeexplore.ieee.org/document/8553750.

[20] *Speech-to-Text Service.* URL: https://learn.microsoft.com/en-GB/azure/ai-services/speech-service/get-started-speech-to-text?tabs=macos%2Cterminal&pivots=programming-language-csharp.

[21] et la. Surabhi Updhyay Shivam Updhyay. *VOICE ASSISTANT FOR VISUALLY IMPAIRED.* URL: https://www.jetir.org/view?paper=JETIRAL06062.

[22] et la. Swati Shilaskar Shivam Ghadge. *English to Marathi Text Translation using Deep learning.* URL: https://ieeexplore.ieee.org/document/9865781.

[23] *Text-to-Speech Service.* URL: https://learn.microsoft.com/en-GB/azure/ai-services/speech-service/get-started-text-to-speech?tabs=macos%2Cterminal&pivots=programming-language-csharp.

[24] *Vision Empower Trust.* URL: https://visionempowertrust.org/.

[25] *What is conversational language understanding?* URL: https://learn.microsoft.com/en-gb/azure/ai-services/language-service/conversational-language-understanding/overview.