ISSN NO: 2249-3034

METAFRASI: An Android Application for Converting Speech to Text in Multiple Regional Languages

Dr.ARVIND PRASAD

Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India

Email: juwairia.sayyed20@vit.edu , hamza.karani20@vit.edu , tejaswini.katale20@vit.edu , adityaraj.honraopatil20@vit.edu , saraswati.jadhav@vit.edu

ABSTRACT

In the present world, communication is the key element to progress. There are about 7099 languages spoken all around the globe. In India alone, the number of spoken languages stand at 454. Speech is one of the oldest and most natural means of information exchange between humans. However, with the existence of so many languages, there develops a communication barrier creating confusion and disrupts meaningful conversation. METAFRASI: An Android Application developed by the authors aims to bridge the communication barrier by converting Speech to Text in multiple regional languages of India.

Keywords— Speech to text; Indian languages; translation; Android Application; Voice Recognition; Signal Processing; API

I. INTRODUCTION

Speech is the basic and most efficient form of communication method for people to interact with each other[1]. In a computer aided speech recognition system, a person speaks through a microphone or telephone and the computer listens which more often means the computer simply attempts to transcribe the speech into a textual form. Speech–to–text research has found new ideas to help differently abled people with voice prompted writing tools[2].

Around 7000+ languages exist around the globe today and there are about 454 languages spoken in India. India has 22 official languages according to its constitution; the most of any country. With so many languages, communicating may be a challenging task if both parties involved in communication do not agree upon the same language. To combat this communication barrier, the authors of this paper propose an android mobile application that will assist in translation of multiple Indian regional languages from speech to text. The application is called Metafrasi, which is a Greek word for 'Translation'. The application is developed using Java and also makes use of Google API for effectively converting said speech to text. The main languages of focus remain Hindi, Tamil, Telugu, and Marathi for development phase. This research paper also aims to observe different techniques and algorithms that are applied to achieve the mentioned functionalities. The system consists of two components, the first component is for processing acoustic signals which are captured by a microphone and the second component is to interpret the processed signal, then mapping the signal to words[3]. This system intends to focus only on the acoustic signal processing without the incorporation of a visual input. The authors also aim to extend the system to incorporate Text to Speech and Speech to Speech functionalities to further bridge the gap of communication.

II. LITERATURE REVIEW

Research conducted in the field of Text to Speech and Speech to Text conversions over the last 30 years has been carried out in various aspects, especially in the field of ICT. Speech recognition systems have already advanced from laboratory to practical applications; there have already been more mature market products.

In the research conducted by Ayushi Trivedi et al[1], various STT and TTS techniques along with their applications are studied. They concluded that HMM works as a better speech signal to text converter and has a good computational feasibility. Formant synthesis that makes use of parallel and cascade synthesis works as the best Text to Speech converter.

S Sultana et al.[4] used Speech Application Program Interface to explore Speech-to-Text conversion for Bangla language. Although the results that they achieved are promising for the respective studies, they identified several elements to improve the performance and accuracy of the proposed system.

Yee-Ling Lu, Man-Wai and Wan-Chi Siu[5] explain about text-to-phoneme conversion by using recurrent neural networks trained with the real time recurrent learning (RTRL) algorithm.

W.D. Lewis[6] used a speech-to-speech pipeline and extended Skype's functionalities to provide a tool for voice communication to aid people with severe hearing disabilities. This was composed of a speech-to-text transcription step, a machine translation engine, and a text-to-speech synthesizer. The results implied this pipeline was powerful enough to run in real-time.

S. Venkateswarlu [7] created a portable TTS device that takes input as text images and converts it to speech with high performance and a readability tolerance of less than 2%, with the average time processing less than three minutes for an A4 paper size. The device used Festival; an open-source TTS system available in multiple languages.

In the research paper "Text-to-speech algorithms based on FFT synthesis"[8] the authors present FFT synthesis algorithms for a French text-to-speech system based on diaphone concatenation. Fast Fourier Transform synthesis techniques are capable of producing high quality prosodic adjustments of natural speech. Various different approaches are explored to reduce the distortions due to diaphone concatenation.

P.S. Nayak[3] developed a Bangla text-to-speech conversion engine in Java that reads and converts Bangla webpage to speech. The algorithm utilized is letter-to-sound rules along with Text Normalization. However, the speech output is monotonous.

To enhance speech output in TTS conversions, Vatsal Aggarwal[9] proposed a model that improves speech naturalness, emotional intensity, and signal quality for which they use Vanilla VAE and VAE+FLOW. They provided a method to perform one-shot adaptation of expressive speech that performed better over Neutral TTS. Their research proved that signal quality is the obstacle that prevents improvement of naturalness and emotional strength.

III. METHODOLOGY

The development of this system was split into three main phases: The Graphical User Interface Design, Command line logic for conversion of speech corpus, and Android Application development.

A – Graphical User Interface Designing

The Graphical User Interface was designed using a designing tool 'Figma'. The primary reason for this is the ability for collaborating online. The design explains what the application will look like and what will be the available features of the proposed system as a whole. It consists of a basic user authentication page, a password reset page, and a frame to allow user to choose between Speech to Text conversion, Text to Speech Conversion and Speech to Speech Conversion. Once clicked on any of these options, the GUI moves ahead onto frames explicitly created for each of these functionalities as shown in figures below.

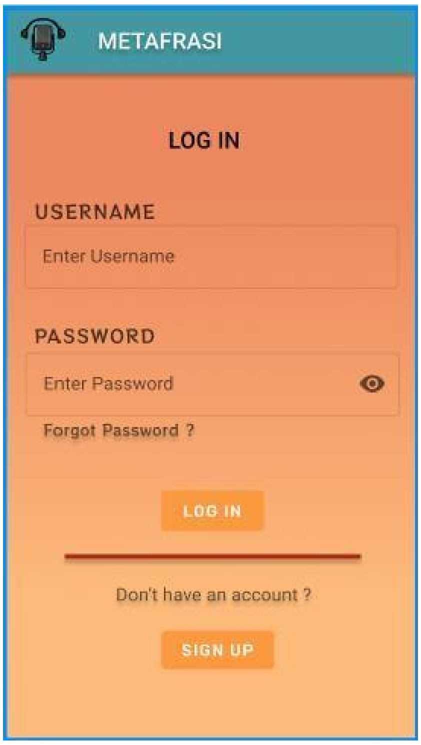

Fig 1.1 Login Page

Figure 1.1 depicts how the application will look like when started. It is the first frame of the app. The user can log into the account by entering the username and password. If the user does not have an account, the sign-up procedure has to be followed by clicking on the sign-up button in this frame which takes the user to, yet another frame as show in Figure 1.2.

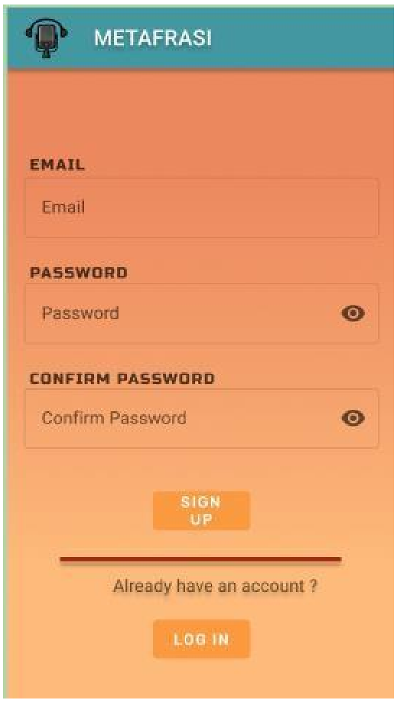

Fig 1.2 Sign Up

In case the user has forgotten the password, a reset password link is sent to the registered email id of the user. The design layout of the reset password is not provided so as to avoid overcrowding of images. After successfully having logged into the account, the user can access the homepage of the mobile application.
Fig 1.3 below shows the homepage of the application. The user can choose one of the three displayed options: Speech to Text, Text to Speech or Speech to Speech Conversion. Once an option is chosen by clicking onto the button, the user is moved to the next respective frame as per the desired option for further input of either audio or text.

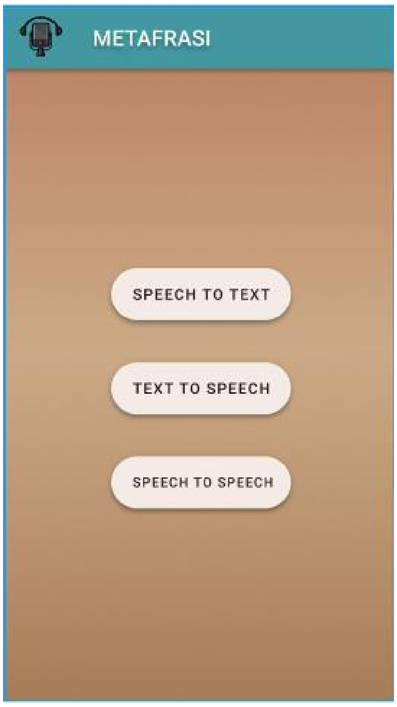

Fig 1.3 Home Page

Figure 1.4 below shows the frame that is displayed to the user when the user clicks on the Speech to Text (STT) button on Homepage frame (Fig 1.3). The user can click on the mic icon to record audio which is processed and converted to the choice of regional language according to the user.

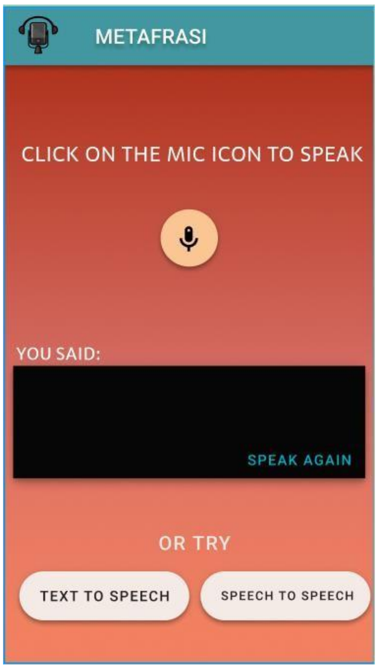

Fig 1.4 Speech to Text frame

For ease of navigation of the users, the options to move the other two frames i.e., Text to Speech (TTS) and Speech to Speech (STS) options are also provided at the bottom of the frame.

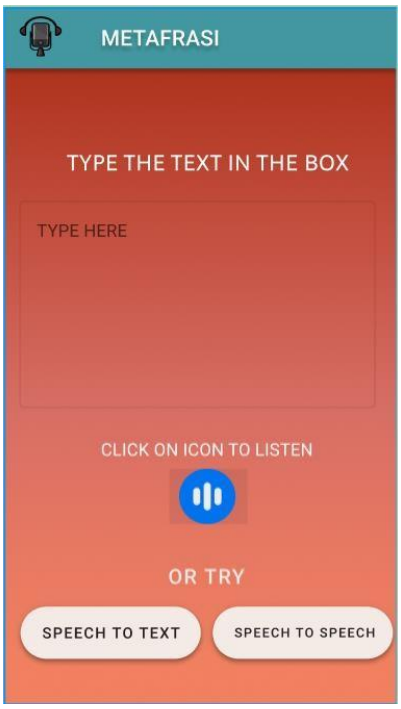

Fig 1.5 Text to Speech frame

Fig 1.5 is a display of the Text to Speech frame where the user inputs a text in any of the regional languages of choice using keyboard and then after processing the text, it is converted into audio of the desired language of translation. The user can click on the icon to hear the audio after conversion.

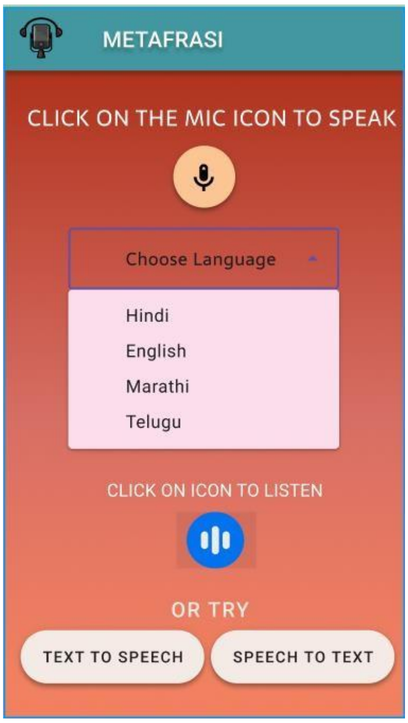

Fig 1.6 Speech to Speech frame

Figure 1.6 above is a display of the Speech-to-Speech frame of the Metafrasi mobile application. The user can click on the mic icon to speak in any of the regional languages. Then, the user chooses the language to translate the spoken audio into. Once the speech corpus has been translated and processed, the user can listen to the audio by clicking on the audio icon.
The above Graphical User Interface is designed using Figma and further development of the android application is carried out on Android Studio in Java language. For effective conversion of Speech to text and Text to Speech, Google API's Speech Recognizer is interfaced with the mobile application for the online version of the application.

B – Command Line Translation using Python

The library gTTS in python is imported so as to interface with Google Translate's Text-to-Speech API. The principle of work is that it writes spoken audio data in MP3 format to a file-like object known as ByteString for further audio manipulation, or standard output(stdout). It features flexible pre-processing and tokenizing. Speech-to-Text has three main methods to recognize speech. Synchronous Recognition, Asynchronous Recognition & Streaming Recognition. Synchronous Recognition sends audio data to the API, performs recognition and processes that data. The recognition requests are limited to audio data of 1 minute or less in duration [10].
Asynchronous Recognition (REST and gRPC) sends audiodata to the Speech-to-Text API and initiates a *Long Running Operation*. Using this operation, users canperiodically poll for recognition results.

Streaming Recognition (gRPC only) performs recognition on audio data provided within a gRPC bi- directional stream[11]. Streaming requests are designed for real-time recognition purposes, such as capturing live audio from a microphone; it provides interim results while audio is being captured, allowing results to appear. In this project, the synchronous recognition requests to perform the conversion from speech to text.

C – Android Application Development

The final phase of the proposed system will be an android application development in Java language using Android Studio as the Integrated Development Environment. The system design will be as per the first phase GUI design as depicted through Fig 1.1 to Fig 1.6. This application will make use of Google API's Speech

Recognizer for the online version i.e., when Wi-Fi service is available and work on the command line developed version using the gTTS library of python for the offline version i.e., when no active Internet connection is available.

IV. RESULTS

The general objective of the proposed system is to develop a Text-to-Speech and Speech to Text synthesizer for the physically impaired and the vocally disturbed individuals using English, Marathi, Hindi and Telugu, Tamil, and Marathi language. The specific objectives are:

1. To enable the deaf and dumb to communicate andcontribute to the growth of an organization through synthesized voice.
2. To enable the blind and elderly people to enjoy a user-friendly computer interface.
3. To implement an isolated whole word speech synthesizer that is capable of converting text and responding with speech and vice versa .
4. To validate the automatic speech synthesizer developed during the study.
5. It recognizes numbers as well.

A – Application Programming Interfaces (APIS)

Enables developers to convert audio to text in over 125 languages and variants, by applying powerful neural network models in an easy-to-use API.

1) TEXT TO SPEECH (OFFLINE VERSION)

The main drawback is that it is restricted to the voices and languages that are pre-built on to the system and are often monotonous.

Pyttsx3: It is a text -to -speech conversion library and unlike other libraries it works in an offline mode.

Functions used:

i) pyttsx3.init([driverName:string, debug: bool]):
Pyttsx is the imported library whose instance is stored in engine variable initialized by engine=pyttsx3.init().It accepts 2 parameters :
Driver name: It loads the best and the current driver available for the platform.

Debug:

1. ImportError: Raises error when the requested driver is unavailable.
2. RunTimeError: Raises error when the driver is unable to initialize.

ii) engine.say(text)
After importing pyttsx library engine.say() function is called which takes 2 parameters "text unicode" and "string"(optional). For text parameter, text is entered which the user wish to hear and the string parameter is used to set a name for the text we want to hear.

iii) engine.runAndWait():
After taking the input from the user "say command" , this function will make it audible to the system.

iii) engine.getProperty(name : string):
As "engine.say()" takes the input in the form of string as a parameter ,it then returns the object matching with the string.

iii) getProperty(name: string): This function gives us the current engine properties.

iv) setProperty(name,value): It changes the current properties which changes the utterances.

2) SPEECH TO TEXT (OFFLINE VERSION)

Recognizer Class: This is the class which helps to convert audio files to text.

i) speech_recognizer:
This package is installed using pip install speechrecognition .It recognizes the speech and converts the spoken words into text with the help of microphone.

ii) Pyaudio :
Pyaudio is used to take the input audios from the user using microphone.

iii) sr.recognizer():
It is the instance created of the speech_recognizer class.

iv) sr.Microphone.list_microphone_names():
This function will return a array/list of the microphones connected to the system.

v) r.adjust_for_ambient_noise(source) :
The microphone is able to detect the audio only when there is silence in the surroundings so this function adjusts the energy threshold dynamically using source's audio.

vi) r.listen(source) :
This function listens to the first phase of the speech and extracts it into audio data files. Also, it gives error when the microphone is unable to hear spoken text.

3) SPEECH TO TEXT (ONLINE VERSION)

i)mic_list = sr.Microphone.list_microphone_names()
This function generates all the audio cards /microphones and so from this list we can select whichever mick we want to use for the program.

ii) text = r.recognize_google(audio)
This function recognizes the audio from google.

iii) sr.UnknownValueError/ sr.UnknownValueError:
If there is background noise near the microphone , in such case google will not able to hear the spoken speech and hence it will generate an error.

Working of TTS API's:

1) Input Text (Raw text or Tagged Data)
2) Engine:

- **Text Analysis: Document Structure Detection**
  This API uses the computer software's to automatically analyze text and also undergoes text mining and extraction. It independently sorts and classifies language detection.
- **Text Normalization**: It is a pre-process of transforming the text into a verbalized form. The 2 main steps are:
  Lemmatization and Stemming: These are the
  Methods used to analyze meaning the work. Stemming used the stem of the word and
  Lemmatization uses the context of the word.
- **Linguistic Analysis: It refers to the analysis of language sample.**
  1)Phonology: It is the analysis of universal and non-universal qualities of sound.
  2)Morphology: It uses prefix and suffix to analyze the word.

3) Phonetic Analysis:

- **Homograph Disambiguation:** It presents a statistical decision for lexical ambiguity.
- **Grapheme-to-Phoneme Conversion:** The text is converted into phonetic transcription using Letter-to-Sound rules. The synthesis of new speech, TTS system selects the recorded phoneme units (PUs) from database and modifies the duration based on the rule based on spelling using (TD-PSOLA).

4) Prosodic Analysis: Prosody refers to duration, intonation and intensity patterns of speech associated to the sequence of syllables, words and phrases. It helps to develop a high-quality speech pattern.

5) Speech Synthesis:

- **Voice Rendering:** It is a process of changing sound effects to experience reality of a situation.

6) Audio is out.

Working of STT API's:

1) API'S: The "REST" and "gRPc" send audio data to API's. It performs recognition on the data and returns processed audio.

2) Engine

- **Acoustic Model**: It establishes a statistical relation between the sound waveform received and linguistic units (HMM-Hidden Markov Model).
- **Lexicon**: It helps to fit an appropriate word on the basis if the speech which is heard.
- **Language Model**: It receives phonemes from the received speech and it uses its learned probabilities to select the right words. It gives the probability of next sequence of words.

3) Decoded Text Output

V. CONCLUSION

The authors investigated various existing models, their techniques and algorithms and proposed a novel approach to combat communication barrier that arises when conversing in an unknown language. Although there is a long way to go to achieve fully seamless, real-time spoken translation, in Metafrasi there is the potential for real-time, open-domain, cross-lingual conversations. The android mobile application developed in Java translates multiple Indian regional languages from speech to text making effective use of Google API for the functionality. The system comprises of two components, one that processes acoustic signals captured by a microphone and the second that interprets the processed signal and maps them to corresponding words. This technology opens doors between communities and differently enabled people facilitating unbridled communication between fellow human beings.

REFERENCES

[1] Ayushi Trivedi, Navya Pant, Pinal Shah, Simran Sonik and Supriya Agrawal, "Speech to text and text to speech recognition systems-A Review", OSR Journal of Computer Engineering (IOSR-JCE)e-ISSN: 2278-0661,p-ISSN:2278-8727, Volume 20, Issue 2, Ver. I (Mar.- Apr. 2018), PP 36-43.

[2] Uliniansyah, Mohammad & Hammam, Riza & Santosa, Agung & Gunarso, Gunarso & Gunawan, Made & Nurfadhilah, Elvira. (2017). Development of text and speech corpus for an Indonesian speech-to-speech translation system. 1-5. 10.1109/ICSDA.2017.8384448.

[3] P. S. Nayak, "Bangla web page reader - An approach to Bangla text-to-speech conversion," International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), 2014, pp. 1-3, doi: 10.1109/ICRAIE.2014.6909266.

[4] Sultana, S., Akhand, M. A. H., Das, P. K., & Rahman, M. H. (2012, July). Bangla Speech-to-Text conversion using SAPI. In 2012 international conference on computer and communication engineering (ICCCE) (pp. 385-390). IEEE.

[5] Yee-Ling Lu, Man-Wai Mak and Wan-Chi Siu, "Application of a fast real time recurrent learning algorithm to text-to-phoneme conversion," Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, pp. 2853-2857 vol.5, doi: 10.1109/ICNN.1995.488186.

[6] Lewis, W.D. (2015). Skype Translator: Breaking down language and hearing barriers. A behind the scenes look at near real-time speech translation. TC.

[7] Venkateswarlu, S. & Duvvuri, Duvvuri B K Kamesh & Jammalamadaka, Sastry & Rani, R.. (2016). Text to Speech Conversion. Indian Journal of Science and Technology. 9. 10.17485/ijst/2016/v9i38/102967.

[8]F. Charpentier and E. Moulines, "Text-to-speech algorithms based on FFT synthesis," ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing, 1988, pp. 667-670 vol.1, doi: 10.1109/ICASSP.1988.196674.

[9] Aggarwal, Vatsal & Cotescu, Marius & Prateek, Nishant & Lorenzo-Trueba, Jaime & Barra-Chicote, Roberto. (2020). Using Vaes and Normalizing Flows for One-Shot Text-To-Speech Synthesis of Expressive Speech. 6179-6183. 10.1109/ICASSP40776.2020.9053678.

[10] R, Reshma. (2020). Speech Recognition using Deep Learning Techniques. International Journal for Research in Applied Science and Engineering Technology. 8. 2199-2201. 10.22214/ijraset.2020.6358.

[11] Rumia Sultana, Rajesh Palit, “DEVELOPMENT OF TEXT AND SPEECH CORPUS FOR AN INDONESIAN SPEECH-TO-SPEECH TRANSLATION SYSTEM”, The 9th International Forum on Strategic Technology (IFOST), October 21-23, 2014, Cox’s Bazar, Bangladesh