A Real Time Speech to Text Conversion Technique for Bengali Language

Abdullah Umar Nasib
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
umarnasib13@gmail.com

Ruhan Ahmed
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
ruhanahmedaqua@gmail.com

Humayun Kabir
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
humayunkabirtorab@gmail.com

Jia Uddin
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
engrjiauddin@gmail.com

***Abstract*— This paper presents a model to convert natural Bengali language to text. The proposed model requires the usage of the open sourced framework Sphinx 4 which is written in Java and provides the required procedural coding tools to develop an acoustic model for a custom language like Bengali. Our main objective was to ensure that the system was adequately trained on a word by word basis from various speakers so that it could recognize new speakers fluently. We used a free digital audio workstation (DAW) called Audacity to manipulate the collected recording data via continuous frequency profiling techniques to reduce the Signal-to-Noise-Ratio (SNR), vocal leveling, normalization and syllable splitting as well as merging which ensure an error free 1:1-word mapping of each utterance with its mirror transcription file text. To evaluate the performance of proposed model, we utilize an audio dataset of recorded speech data from 10 individual speakers consisting of both males and females using custom transcript files that we wrote. Experimental results demonstrate that the proposed model exhibits average 71.7% accuracy for our tested dataset.**

Keywords*— *Text Recognition; UNICODE; CMU Sphinx; Digital Audio Workstation; Speech-to-Text; Real-Time Conversion

I. INTRODUCTION

Bengali is one of the richest languages in the world and about 250 million people all over the world speak the language. However, typing in Bengali can be difficult especially for those who cannot type fast. Being able to convert the spoken word to text [1, 2] is the easiest form of typing any language and we want Bengali to be a part of that digital world. There are regional variations in Bengali Language called dialect. Style of pronunciation and accent are different from area to area. Some words are even pronounced differently by different speakers [7].

In the past, contributions in the exact field were achieved using Microsoft SAPI but it was very limited in being slow in recognizing continuous speech without proper word gap [5]. We hope to achieve faster performance using Sphinx 4. The reason that the Bengali language has fallen behind in this regard is because of being a very tricky language with a lot of challenges to overcome to do correctly [4]. Speech to text conversion is the process of dissecting discrete syllables or phonemes of recorded vocal audio and converting them to their literal transliteration [6]. Moreover, one of the main challenges previous works suffered to convert voice into text was developing the phonetic dictionary [8] which is used to match portions of the audio to their respective phonemes [10] that get merged into syllables to finally form the desired word.

On the other hand, the purpose of our approach is to use a modern engine like the CMU Sphinx 4 [9] that allows more functionality than old engines like SAPI to deliver faster continuous speech recognition via improved algorithms. As Sphinx was designed for English based letters our key approach is to convert the spoken Bengali speech into the modern widely renowned "Banglish" form which essentially Bengali is written with English characters. We wrote our own scripts in Java to perform this on the data sets we wrote. After the language model and dictionary are generated, we replace the "Banglish" words back to Bengali. We also built a script to handle Bengali spell checking so that only a single spelling instance of a word exists. In this paper, we propose a technique that will convert Bengali speech into text in real time. Through microphone this system takes Bengali voice as input. The voice input passes through different phases into the system and starts processing. Hence, completing all the checking if the input is error free and mapped correctly, the given voice input gets converted into Bengali UNICODE text.

The rest of the paper is organized in the following sequence. Section II discusses the proposed model with a described workflow; block diagrams with details elaboration whereas testing the technique with different sets of data model and the results are described in Section III. This section also discloses the methods of analyzing experimental result of the proposed technique. Finally, section IV concludes the paper.

II. Proposed Model

This section presents the detail descriptions of proposed model which contains several phases. Fig. 1 represents the whole working procedure of our proposed model.

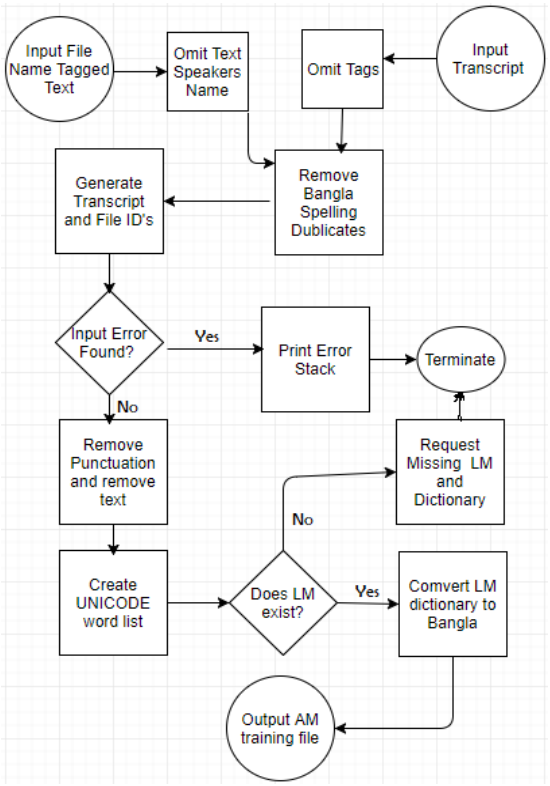

Fig. 1. Working Flow Chart

In our proposed technique, we can divide the whole system into two different segments named 'Training' and 'Testing'. Firstly, in the training segment, we trained our system using a number of modules which are shown in Table 1. The technique takes text input using two different methods. One is 'Transcript' which is showed in Fig. 2 and 'file name tagged text' is the other which is in the format "<<<SpeakerName>>> Text <<AudioName>>" that parses an input text to collect the respected speaker name and audio name for the spoken text to train. We interchanged between the two based on our needs.

The way we input transcripts is, the tags such as <s> ডাল </s> (nasib1w3) where 'ডাল' is tagged with the speaker name 'nasib', script line number 1 and word number 3 gets removed and stored in this phase shown in Fig. 2. Before that, when we took the audio input as data set for training we had to divide the vocal recordings into words. Based on the silence between two words when pronounce, we separated the words from a continuous spoken sentence.

Therefore, splitting the wave in parts to train the system was the first phase, shown in Fig. 5. The transcript may have same words of different spelling which may create problems. Solving this situation, the next phase which is 'Remove Bengali Spelling Duplicates' mentioned in Fig. 2 was applied. This phase removes the duplicate words and keeps only unique words in the transcript. Using the algorithm-1 the duplicate words are being detected.

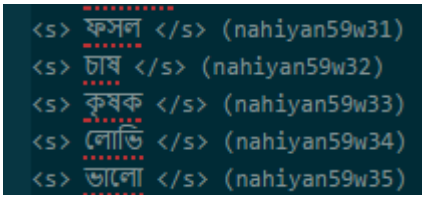

Fig. 2. Tagged input transcript

Algorithm-1: Unique UTF Word Finder Algorithm

```
List of Bengali Tokens is input that contains
duplicates
For Each Bengali Token
  Convert to Bengali Phonetic English
  Add to Banglish Token List
End
  Create Unique and Duplicate Hash Sets
For Each Banglish Token
  If it does not exist in Unique List
     Add to Banglish Unique List
  Else
     Add to Banglish Duplicate List
End
For Each Banglish Duplicate
  Create new Lines Integer List
  For Each Banglish Token
     If Banglish Token Equals Banglish
Duplicate
     Add iteration value to Lines List
  End
  Create new Bengali Duplicate List
  For Each Lines value
     Get Bengali Tokens value at position
[Lines value]
     Add the returned value to Bengali
Duplicate List
  End
  For Each Bengali Token
  For Each Bengali Duplicate
     If Bengali Token Equals Bengali Duplicate
        Set this Bengali Token to first value
of Bengali Duplicate
     End
  End
End
Return new Bengali Tokens List
```

Generating transcript is the most important phase as the transcript will be used to train the system. It also generates specific ID for every individual word. Then we check whether there is any existing error in the input transcript. If any error is detected, the error stack which keeps the record of errors is printed. Otherwise we use the Algorithm-2 to convert Bengali to "Banglish", which is the phonetic form of Bengali written in English.

Algorithm-2: Bengali UNICODE to Bengali Phonetic English Algorithm

```
Create Bengali Token and Banglish Token List
```

```
For Each Bengali Token
Remove Special Characters that are not needed
        Fix Special Exceptions in Certain
UNICODE Characters
        Cross Reference Character Sequence
with Stored
        Phonetic English Equivalent
        Replace Matching Character Sequences
        Add to Banglish Token
End
Return Banglish Token
```

After running this algorithm, punctuations like ◌ঃ, ◌ঁ, ৎ, ◌ঁ are also removed to reduce the complexity. Then the transcript generates unique word list. Then we check for the existence of Language Model (LM) [11]; If found, the file generation is completed. Otherwise the technique will request it [12] and following that will get terminated.

Finally, the 'Training' segment ends with the last phase where the Acoustic Model (AM) gets generated. The next segment 'Testing' begins from here.

AE	K	EH	OY
CH	SH	ER	S
IY	L	G	AW
D	T	AY	TH
AH	OW	P	F
R	AA	Z	AO
IH	UW	B	V
N	M	HH	DH
JH	EY	Y	SIL

Fig. 3. Phone Set

The phoneme list in Fig. 3 has been used to make the words recognizable to the system. Furthermore, to minimize the error rate of the variety of pronunciations, we compare the waves of the input of different speakers for same words. Fig. 4 shows the difference of the same uttered two words for 6 speakers. Running our algorithm, we create text data set for each speaker which is then trained by the system separately.

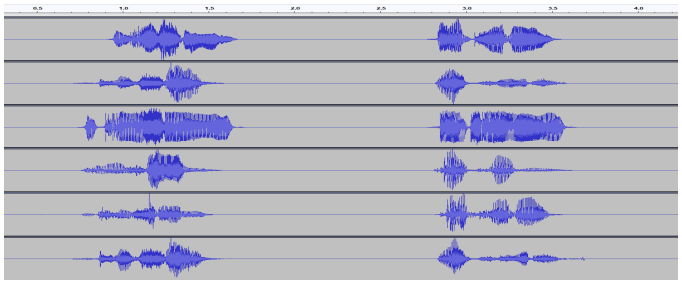

Fig. 4. Vocal variation Training

As Fig. 5 shows, we split each input voice file into separate words. For example, if input voice is 'আমি হাত দিয়ে ভায় খাই', it will be split into 'আমি', 'হাত', 'দিয়ে', 'ভাত', 'খাই'. After that the work of Fig. 1 is executed.

Fig. 5. Word Mapping

Here, the training process of our technique ends and the system is prepared to run with different dataset. In the next segment III, the experimental setup and the result is discussed in details.

III. Experimental Setup and Result

The second segment of our proposed model was testing and in this phase, we tested our system and analyzed the results. We required at least 5 hours (300 minutes) of data but we could not acquire it entirely. We collected the recorded audio of a book of one of the renowned writers in Bangladesh. But we faced problems there as it was not spoken in natural language. Then we chose to make a dataset of our recorded voice. In the experimental evaluation, we experienced better accuracy but another issue arises which was variation of voice. The system was still unable to detect new speakers correctly. Finally, we trained our system with 5 different speakers including male-female, young-old. With 503 unique words of 1.99 hours which gave the best accuracy. Hence, the three dataset and the accuracy output are listed in Table 1.

TABLE I. Classification of Model Data

Model	Model A	Model B	Model C
No. of Speaker	01	02	05
Time(in mins)	58	80	119
No. of Words	371	503	503
M-F Ratio	1:0	1:1	3:2

1. Model A: Dataset of single speaker containing 58 minutes and 371 unique words of one male speaker.
2. Model B: Dataset of double speakers of 80 minutes 503 unique words of one male and one female speaker.
3. Model C: Dataset of five speakers containing 119 minutes and 503 unique words including 2 female and 3 male speakers.

In experimenting with the book recording in Model A the accuracy rate was very poor, 33.6% to be exact.

Testing with Model B containing the voice of one male & one male speaker of 80 minutes gave a better accuracy. The experimental result was 52.3% which is much better than Model A shown in Fig. 6. Still the low amount of data was decreasing the accuracy rate.

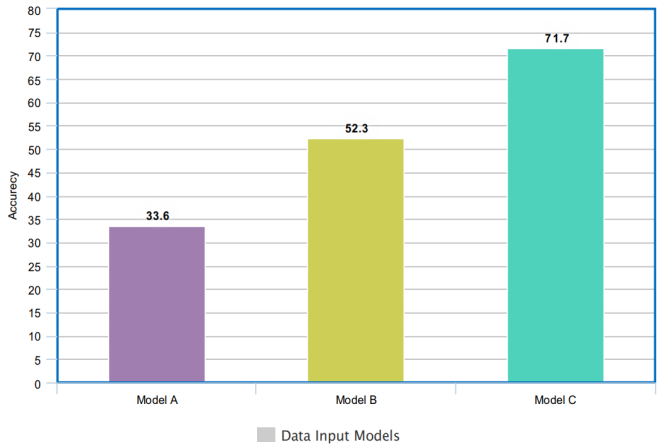

Fig. 6. Comparing the testing result

Finally, using our third dataset model C, having 503 words of 5 speakers and almost 2 hours in length, we managed to get the best accuracy of 71.7%, as shown in Figure 6.

Java program was used to calculate the accuracy which checks if given input word is 1:1 mapped with output word and also if the content of each word matches or not. While testing, we knew about the content of the sentences uttered in the recorded wave files. We wrote it down in a text file with its respective audio file name tagged at the end of each sentence. Then we ran to go through the entire 300 audio files to save each detection text output in a different text file.

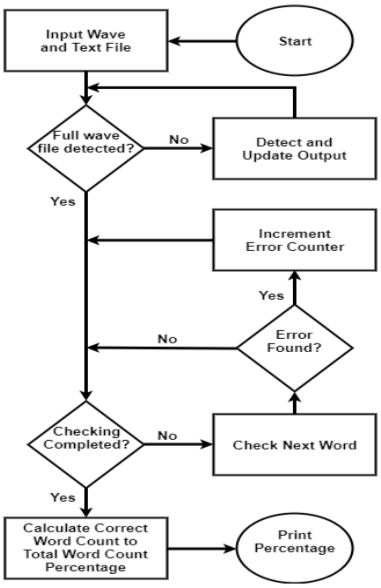

Fig. 7. Accuracy Calculation Flowchart

After finished detecting all the files, we ran our accuracy test script to check the accuracy. This script matched each word from the audio detection text file with each word that we wrote manually knowing what was actually uttered in the audio. If any word was found mismatch, we incremented the error rate variable. After the process completes, we output a single percentage of correct words detected to total number of words. The Fig. 7 shows this entire process in the form of a flowchart.

IV. Conclusion

In this paper presented a model of voice to text conversion method for Bengali language. An open source frame work called Sphinx4 was used to generate Bengali UNICODE font. A digital audio workstation, Audacity was used to manipulate the recorded data. The performance of proposed model was tested using a dataset where both male and female voice was recorded. The proposed model showed 71.7% accuracy for the tested dataset.

References

[1] Prachi Khilari, Bhope V. P., "A Review on Speech to Text Conversion Methods," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 7, July 2015.

[2] B. Raghavendhar Reddy, E. Mahender, "Speech to Text Conversion using Android Platform," B. Raghavendhar Reddy, E. Mahender / International Journal of Engineering Research and Applications (IJERA) vol. 3, Issue 1, January -February 2013, pp.253-258.

[3] Preeti Saini, Parneet Kaur, "Automatic Speech Recognition: A Review," International Journal of Engineering Trends and Technology- Vol.4, Issue.2- 2013.

[4] Mumit Khan, Md. Abul Hasnat, Jabir Mowla, "Isolated And Continuous Bangla Speech Recognition: Implementation, Performance And Application Perspective," Department Of Computer Science And Engineering, Conference Papers (Centre For Research On Bangla Language Processing), BRAC University, 2007.

[5] SULTANA, S. AKHAND, M. A. H., DAS, P. K. & RAHMAN, M. M. H. (2012) Bangla Speech-to-Text Conversion using SAPI. In Computer and Communication Engineering (ICCCE), 2012 International Conference. Kuala Lumpur, 3-5 July 2012. Kuala Lumpur: IEEE. P.385-3.

[6] Md. Farukuzzaman Khan, Md. Mijanur Rahman, Mohammad Ali Moni (Pust), "Speech Recognition Front-End For Segmenting and Clustering Continuous Bangla Speech," Daffodil International University Journal of Science and Technology, Vol 5, Issue 1, 2010.

[7] Firoj Alam, Mumit Khan, S.M. Murtoza Habib, "Bangla Text To Speech Using Festival," Conference on Human Language Technology for Development, Alexandria, Egypt, May 2011.

[8] Mir Ashraf Uddin, Nazmus Sakib, Esrat Farjana Rupu, Md. Afzal Hossain, Md. Nurul Huda, "Phoneme based Bangla text to speech conversion," 2015 18th International Conference on Computer and Information Technology (ICCIT), 2015. P.531 – 533.

[9] K. M Shivakumar, K. G. Aravind, T. V. Anoop, Deepa Gupta, "Kannada speech to text conversion using CMU Sphinx," 2016 International Conference on Inventive Computation Technologies (ICICT), Vol 3, 2016.

[10] Suniti Kumar Chatterji, "Bengali phonetics," Bulletin of the School of Oriental Studies, University of London, vol.2, no. 1, pp. 1-25, 1921.

[11] Ghulam Muhammad, Mohammad Nurul Huda, Manoj Banik, Bernd J. Kroger, "Phoneme recognition based on distinctive phonetic features (DPFs) incorporating a syllable based language model", 2009 12th International Conference on Computers and Information Technology, 2009. P.285 – 289.

[12] M. A. Zissman, "Language identification using phoneme recognition and phonotactic language modeling," 1995 International Conference on Acoustics, Speech, and Signal Processing, vol.5, P.3503 – 3506.

[13] Matthew Fifer, Nathan Crone, Griffin Milsap, Nitish Thakor, "Listening to the music of the brain: Live analysis of ECoG recordings using digital audio workstation software," 2013 6th International IEEE/EMBS Conference on Neural Engineering (NER), 2013, P.682 – 685.