HMM Based Kannada Speech Synthesis using Festvox

Sadashiva V Chakrasali, K Indira, Shashank B Sharma, Srinivas N M , Varun S S

Abstract*—*The process which involves generation of human like voice by a machine is called speech synthesis. The developments in the fteld of speech synthesis is vast in international languages, but it is limited in Indian languages like Kannada. This work aims at development of such a system for Kannada language using Festival and Festvox. It is based on parametric analysis and models of speech features, particular to a language and speaker. The system is memoryless and dynamic, wherein only extracted features are stored but not recorded audio. The training process involves speech data acquisition, pre-processing, labelling using Baum-Welch Iteration, whereas testing process involves text analysis, text segmentation, speech synthesis and quality enhancement using acoustic HMM model development. The quality of synthesis is 3.52 dB to 5.02 dB as measured by Mel-Cepstral Distortion (MCD) score.

Index Terms*—*Baum-Welch algorithm, Hidden Markov Model (HMM), Kannada, MCD score, Speech synthesis.

I. INTRODUCTION

SPEECH is a vocal communication process, based on a particular language. A language is a set of specifically articulated vowels and consonants that can produce all words in that language. There are thousands of actively spoken languages across the world. Though a language has a well defined set of phonetic combinations with particular pronunciation, it can vary depending on region, influence of other languages, etc.

A text to speech (TTS) system is a speech generation method for the given text input. A TTS system must be capable of producing synthetic voice to all phonetic combinations in the language. The development of TTS system is mainly application specific. A specific procedure followed to obtain a TTS application, might result in desired outcome for that application. The trade-off between the different types of words to be synthesized and quality of synthesized speech, has to be taken into account while deciding on the approach to follow.

The approach we follow is developed based on the features of speech is considered as parametric TTS. There is no hard coding procedure to synthesise speech in this method, while it involves extracting parameters from speech dataset to build an acoustic speech model. The process involves identifying and obtaining features, training and excitation to obtain synthesised speech. It is a memory less system and consumes minimal resources.

This work involves the development of a parametric TTS using Festival toolkit for Kannada.

II. TOOLKITS AND LIBRARIES

The speech processing toolkits that are used in the implementation are mentioned below

A. Edinburgh Speech Tools (EST)

The Edinburgh Speech Tools Library is a collection of C++ class, functions and related programs for manipulating the sorts of objects used in speech processing. It includes support for reading and writing waveforms, parameter files such as LPC, Cepstra, f0 (fundamental frequecy of a speaker, also known as pitch) in various formats and converting between them.

B. Speech Processing Toolkit (SPTK)

The Speech Signal Processing Toolkit (SPTK) is a suite of speech signal processing tools for UNIX environments, e.g., LPC analysis, PARCOR analysis, LSP analysis, PARCOR synthesis filter, LSP synthesis filter, vector quantization techniques, and other extended versions of them. This software is released under the Modified BSD license.

C. Festival Speech Synthesis System

Festival, by University of Edinburg, offers a general framework for building speech synthesis systems as well as including examples of various modules. As a whole it offers full text to speech through a number of APIs (Application Program Interface) from shell level, though a Scheme Command Interpreter(SCI), as a C++ library, from Java.

D. Festvox by CMU Speech Group

Festvox project is part of the work at Carnegie Mellon University's (CMU) speech group aimed at advancing the state of Speech Synthesis. The Festvox project aims to make the building of new synthetic voices more systemic and better documented, making it possible for anyone to build a new voice. The documentation, tools and dependent software are all free without restriction (commercial or otherwise).

Manuscript published on 30 September 2019
* Correspondence Author

Sadashiva V Chakrasali*, Department Of Electronics And Communication, Ramaiah Institute Of Technology Bangalore, India
K Indira, Department Of Electronics And Communication, Ramaiah Institute Of Technology Bangalore, India
Shashank B Sharma, Department Of Electronics And Communication, Ramaiah Institute Of Technology Bangalore, India
Srinivas N M, Department Of Electronics And Communication, Ramaiah Institute Of Technology Bangalore, India
Varun S S, Department Of Electronics And Communication,Ramaiah Institute Of Technology Bangalore, India

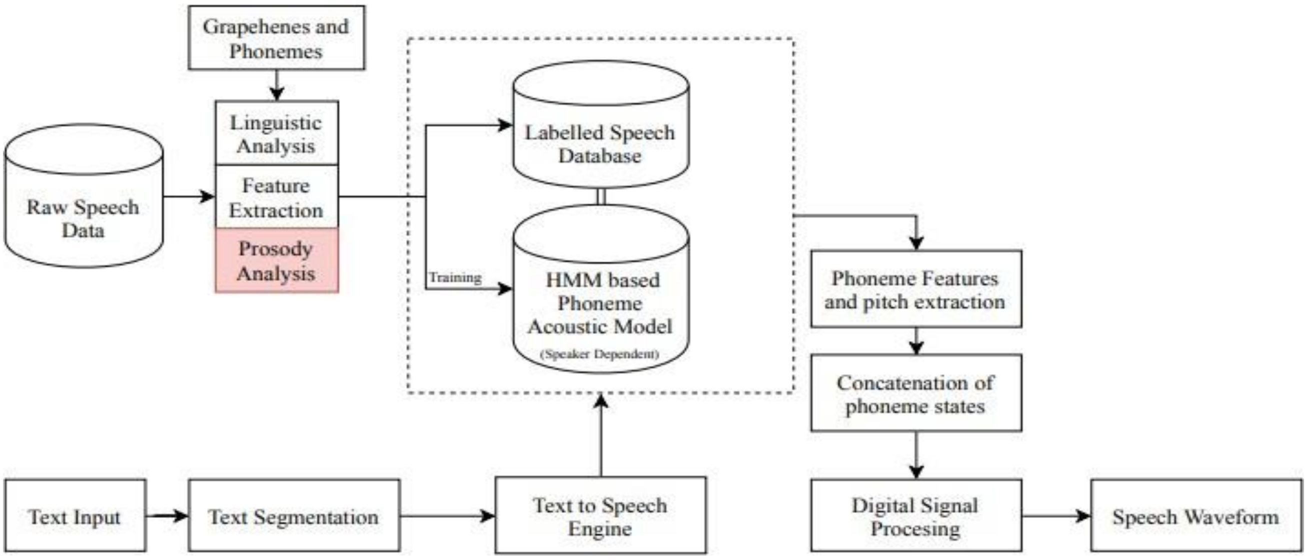

Figure 1. Block diagram of Speech synthesis system

III. PROPOSED METHODOLOGY

The system consists of frontend and backend systems where the frontend deals with user interface for accepting text input and providing synthesized output while the backend deals with various scripts, models, toolkits and database which interact with each other to provide required result.

The linearly connected rectangular blocks in bottom layer of the block diagram describes the frontend portion. The blocks above them describe the backend processes consisting of the raw database for training speech model, collection of graphemes and phonemes, HMM model, labeled database and extracted speech feature files. The frontend accepts user input in the form of Kannada text, which is segmented and fed as an input to TTS engine which communicates with the backend database and trained model to generate raw audio files. Then, post synthesis procedures involving DSP operations like tempo change and pitch shift are performed on the raw audio files to enhance quality. The GUI developed as a part of frontend plays the synthesised speech.

The backend consists of various speech processing toolkits viz. EST, Festvox, Festival and SPTK which are used for processing the speech data at different stages. It also has speech data collected from one female speaker which has both audio files and corresponding labeled textfile (txt.done.data) with Kannada unicode mappings. The data is preprocessed (as explained in next section) so that it is suitable for linguistic analysis and feature extraction. Further, data labeling is performed by Baum-Welch iterative method so that the duration

of phonemes and corresponding labels are in

synchronization.

The duration model for the given input text built, is trained using Festvox toolkit to obtain an acoustic model, which provides best mean variance model for each phoneme in the training dataset. The segmented text data is applied to the acoustic model which provides phoneme features in different states. This process involves phoneme generation that is done by exciting MLSA (Mel log spectral approximation) filter using source. The digital signal processing techniques involving noise re- duction and pitch shift algorithms are applied to get human like voice. The processed speech waveforms are sent to frontend, to be played by the user.

IV. IMPLEMENTATION

A. Speech Data Pre-processing

Speech data of about 696 audio files along with the transcripts that is used for training is acquired from CMU dataset. These files are noise free with the sampling frequency of 16kHz which do not need preprocessing operations like Electronic Noise removal, resampling for the purpose of processing. Some of the operations that are necessary are mentioned below.

1) Dynamic Range Compression*:* Dynamic range refers to the ratio of the maximum amplitude to minimum amplitude each of which is a notable contributor to some feature. Dynamic range compression(DRC) or simply compression is an audio signal processing operation that reduces the volume of loud sounds or amplifies quiet sounds thus reducing or compressing the dynamic range of the audio signal. EST audio manipulation

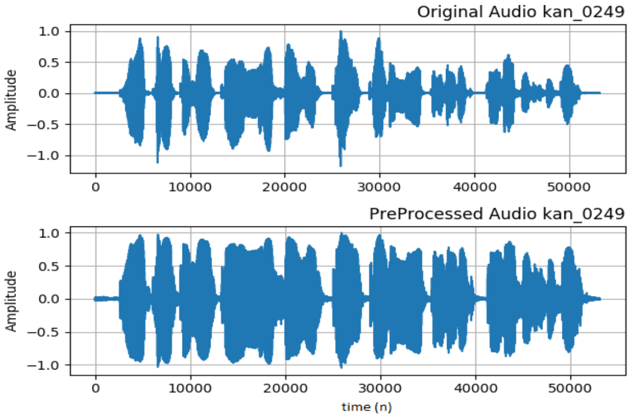

Figure 2. Comparison of Original and Preprocessed audio signal

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

scripts are used to perform dynamic range compression. Fig 2 is a comparison of original audio file and pre-processed audio file after performing the above two operations.

2) ***Silence Pruning/Removal:*** **The** recorded/acquired au- dio signal contains silence at different positions such as be- ginning of signal, in between the words of a sentence, end of signal etc. If silent frames are included, modelling resources are spent on parts of the signal which do not contribute to the identification. The silence present must be removed (above a threshold) or the extent of silence must be reduced before further processing.

Pitch detection algorithm(PDA) is used to determine the start and end of a signal. EST super resolution PDA that produces the fundamental frequency contour from a .wav file can be used to prune silences at the ends and middle.

Two parameters namely silence margin and silence threshold are set while pruning silences. Silence margin (0.1s for intra- sentence and 0.2s for end silence) refers to the number of seconds of silence that the algorithm must retain. The algorithm will crop some part of the silence when the silence is longer than Silence threshold(0.4s).

B. Labelling and Baum-Welch Iteration

Labelling is one of the most important step in building a TTS system. It is the process of aligning phoneme translation with the exact duration at which it occurs in the wav file. The timing information can be any one of start, end or duration of the phoneme.

The file storing labelling data is called a label file with an extension .lab. A similar file which carry more information is an utterance file (.utt) . Utterance files contain information like duration factor, stress level, labels, utterances sorted in id format.It also contains relation token, syllable relations, word relations, and segment relations in tree format. These utterance files can also be used in building a voice using HTS (HMM based Speech synthesis system).

1) Baum-Welch Algorithm*:* This algorithm is used to au- tomatically estimate parameters of an HMM also known as the Forward-Backward algorithm. It is a special case of Expectation Maximization (E.M) algorithm.

1) Start with initial probability estimates.
2) Compute expectations of how often each transition or emission is used
3) Re-estimate the probabilities based on those expectations
4) Above step is repeated until it converges

The labelled data after Baum-Welch Iteration is shown in Table 1. A dummy labelled file is used for initial condition of this iteration.

Table I

LABELLING DATA FOR THE WORD (SHIVARAJ)

	After	Labels
Before	Baum Welch	
Baum Welch(Approximation)		
0.1100	0.08	pau
0.2200	0.18	c}
0.3300	0.245	i
0.4400	0.295	v
0.5500	0.335	A
0.6600	0.35	9r
0.7700	0.455	A:
0.8800	0.485	**J**

C. parameter based acoustic model training and speech synthesis

Each phoneme is represented by five HMM states including two terminating silence/pause states. The statenames are rep- resented by English phoneme equalents with postfix number indicating one of three states, excluding silence states. For ex- ample, the letter ಕ (ka) is represented as k_1, k_2 and k_3. These statenames are basis for representation of phoneme attributes in all label, utterance and model files.

1) *Feature Representation:* The occurrence of each Kan- nada letter gives rise to three feature files as each indic alphabet is considered as a composition of 3 different states. This pro- vides a method for analyzing the effect of adjacent state on a current state. The number of entries in every feature file is equal to the number of occurrences in .wav files.

2) *Classification and Regression Trees:* CART is a method to represent dependency of parameters while making a decision, to achieve a result. The classification involves improvising decision points at the intermediate nodes of the tree while regression involves a method to arrive at best possible result based on nodes points. The leaf nodes of CART specify the output variables/ end-points.

The CART framework for speech synthesis is developed in Festival framework. The training involves building three types of CART models essential for speech synthesis,

3) *Spectral/MCEP CART tree*: To obtain statistics of MFCC relations between phonemesf0 CART tree: To predict pitch at start, middle and end of syllable

Festival uses a statistical method of constructing the above trees. Each of the trees have predefined standard structure used to im-

plement all possible statenames of phonemes. Speech synthesis

involves usage of these CART models

D. *Post-processing*

The quality of synthesized audio file is acceptable as de- scribed by its Mel Cepstral Distortion(MCD) score, obtained by comparing the original audio files available during training with the synthesis audio file of same text generated from the devel- oped HMM.

But the observations made from the synthesized speech files reveal the fact that the fundamental frequency is shifted as compared to training data for the same female speaker. This reduced the clarity in synthesized waves while still the words spoken were clearly understandable.

1) **Pitch Shift:** The sensation of frequency as perceived by human ear is commonly known as pitch of a audio signal. Though pitch is not exactly the frequency of audio, it is the attribute that makes every speaker identical. Pitch is the fun- damental frequency of disturbances in a medium caused by the audio signal, which can be detected by human ear.

The pitch of sounds produced by adult males vary around 90- 180 Hz while that of adult females is around 160-250 Hz. This clear distinction in pitch is the basis of gender identification based on voice.

The shift in pitch as per the speaker's fundamental frequency is applied, which does not introduce change in duration of audio(i.e. the tempo is unchanged). With the known ranges of pitch for male and female voices, it is clear that a female voice needs a positive shift in pitch thereby increasing the difference from that of a male voice.

2) **Tempo change**: Tempo of an audio system refers to the speed at which it is played. The synthesized audio is a combina- tion of phonetic features combined together, where there is high possibility of actual tempo of speaker being lost.

The tempo is usually changed when there is a need to vary the (Beats Per Minute) rate of the audio file. The process of varying tempo introduces a change in duration of audio file as it is stretching or compressing the time scale without changing amplitude.

3) **Noise Removal**: SOX, a sound processing library is used to remove noise. The process consists of developing a noise profile i.e. FFT of the selected noise waveform and cleaning the audio by using the noise profile. The noise profile is dynamic (i.e. 0-0.1s of synthesized audio) and is used to perform noise removal, which improves the understand-ability by a notable amount.

V. TESTING AND QUALITY MEASURES

A. Mel-Cepstral Distortion (MCD Score)

The MCD score is a method to comparatively study the qual- ity of synthesized audio. It calculates a logarithmic deviation of

$$MCD = \frac{10}{ln10\sqrt{2\sum_d \left(mc_{(d)}^{(t)} - mc_{(d)}^{(e)}\right)^2}} \qquad (1)$$

mc(t)(d) : mel-cepstral parameter of training file

mc(e(d)) : mel-cepstral parameter of synthesised file d : index of mel-cepstral parameter array

The value of MCD between 4.5 and 6 dB is considered to be a good result. The synthesis of better quality implies lower MCD score. This work provided MCD score in the range 3.52 to 5.02 dB which proves that the synthesis is natural and intelligible with speaker voice.

B. Mean Spectral Error

A plot of difference between the spectral co-efficients be- tween the synthesized and original audio is shown in fig 3. Here, it is evident that the error in spectral difference is very minimal, indicating similarity in audio files and hence showing the quality of the synthesized audio files

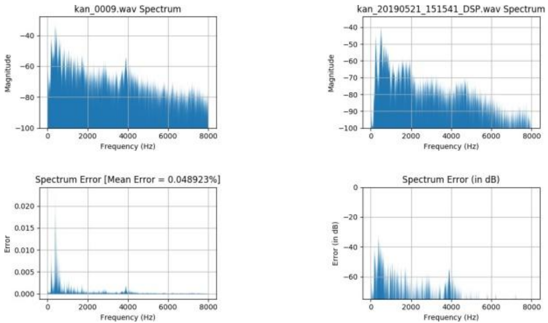

Figure 3. Comparison of spectrum in Original and synthesized audio signal

C. Spectral Ratio slope

This involves plotting a best fit line to the deviation of spectral values and calculating its slope as shown in Fig 4. The slope is ideally zero for two spectra of similar characteristics. Here, we find that the slope is in the range of $\mathbf{10^{-6}}$ which is negligible.

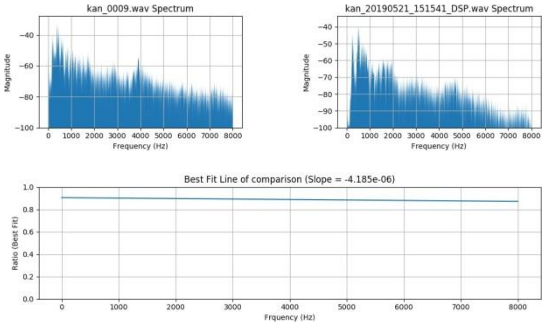

Figure 4. Plot of Spectral variation slope

V. CONCLUSION

The outcomes that were drawn from the work are as follows

1) The speech model is memoryless, which emphasizes that trained features are used to dynamically synthesize text, unlike unit selection where speech for each word is stored.
2) The audio files and their transcripts that were used in training and testing was taken from cmu database and this data was split into two parts, out of which 90% of the data was used in training and 10% of the data was used in testing.
3) The post synthesis processing includes pitch shifting, tempo variation, resampling and noise reduction which enhances the quality of the synthesized audio file.
4) Mel cepstral distortion (MCD) score values, to measure the quality of synthesized audio file, have been recorded. The values are in the range 3.52 to 5.02 which signifies that the audio file is of good quality.
5) The quality of the synthesized audio depends on the train- ing data.
6) The size of the core project files is reduced with the intention of making project deployable. The core project is reduced to 3MB[1]
7) The project is tested on Ubuntu WSL (Windows sub- system for Linux), LUbuntu successfully. The project was also tested on various linux distributions like Debian Linux, Fedora and CentOS where it failed to synthesize audio due to the absence of certain libraries.

The application can be used in real time applications like blind assistance, public announcements etc

VI. ACKNOWLEDGMENTS

We express our deepest gratitude to the developers of various speech toolkits and frameworks which are used, including the CMU Speech Software and CMU speech group.

REFERENCES

1. Mohammed Waseem, C.N Sujatha “Speech Synthesis System for Indian Accent using Festvox”, International

[1]for direct implementation through Festival CLI

2. Journal of Scientific Engineering and Technology Re- search, Vol.03, Issue.34, November-2014
3. Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Ma-suko, Alan W Black and Keiichi Tokuda, “The HMM-based Speech Synthesis System (HTS) Version 2.0”
4. Atish Shankar Ghone, Rachana Nerpagar, Pranaw Ku- mar, Arun Baby, Aswin Shanmugam, Sasikumar M, Hema A Murthy, TBT(Toolkit to Build TTS): “A High Performance Framework to build Multiple Language HTS Voice”, Researchgate conference paper, August 2017
5. Sangramsing N. Kayte, Monica Mundada and Jayesh Gu- jrath, “HiddenMarkov Model based Speech Synthesis: A Review”, International Journal of Computer Applications (0975 – 8887). Volume 130 – No.3, November 2015
6. Sangramsing Kayte, Monica Mundada, Charansing Kayte “Performance Calculation of Speech Synthesis Methods for Hindi language”, IOSR Journal of VLSI and Signal Processing, November-December 2015
7. Festvox Project by CMU speech group,http://festvox.or g/
8. Festival Speech Synthesis system, http://www.cstr.ed.ac. uk/projects/festival/
   Speech Processing Toolkit (Reference Manual), http://sp-tk.sourceforge.net/
9. www.cstr.ed.ac.uk/projects/speech_tools/

AUTHORS PROFILE

Sadashiva V Chakrasali, is currently working as .as a assistant professor in E & C department department, M S Ramaiah Institute of Technology Technology, Bangalore. His research inte interests includes speech modeling, synthesis and machine learning techniques for signal processing.

Dr. K. Indira, is currently working as a Professor in E & C Dept., M S Ramaiah Ins Institute of Technology, Bangalore. Her Research interests are in the field of Image and speech processing.