INTERSPEECH 2017: Show & Tell Contribution
August 20–24, 2017, Stockholm, Sweden

TBT (Toolkit to Build TTS): A High Performance Framework to build Multiple Language HTS Voice

Atish Shankar Ghone[1], *Rachana Nerpagar*[1], *Pranaw Kumar*[1], *Arun Baby*[2], *Aswin Shanmugam*[2], *Sasikumar M*[1], *Hema A Murthy*[2]

[1]Centre for Development of Advanced Computing, Mumbai

[2]Indian Institute of Technology, Madras

{atish,rachana,pranaw}@cdac.in,arunbaby@cse.iitm.ac.in,aswin@jhu.edu, sasi@cdac.in,hema@cse.iitm.ac.in

Abstract

With the development of high quality TTS systems, application area of synthetic speech is increasing rapidly. Beyond the communication aids for the visually impaired and vocally handicap, TTS voices are being used in various educational, telecommunication and multimedia applications. All around the world people are trying to build TTS voice for their regional languages. TTS voice building requires a number of steps to follow and involves use of multiple tools, which makes it time consuming, tedious and perplexing to a user. This paper describes a Toolkit developed for HMM-based TTS voice building that makes the process much easier and handy. The toolkit uses all required tools, viz. HTS, Festival, Festvox, Hybrid Segmentation Tool, etc. and handles each and every step starting from phone set creation, then prompt generation, hybrid segmentation, F0 range finding, voice building, and finally putting the built voice into Synthesis framework. Wherever possible it does parallel processing to reduce time. It saves manual effort and time to a large extent and enable a person to build TTS voice very easily. This toolkit is made available under Open Source license.

Index Terms: speech synthesis, text to speech, hts, Indian language tts

1. Introduction

A Text-to-Speech synthesis system (TTS) converts text into human-like speech which is called synthesized speech. Among many, the two prominent approaches to build TTS systems are: 1. Unit Selection Synthesis (USS) and, 2 .HMM based Statistical Parametric Synthesis (HTS). HTS voice is having several advantages over USS voice e.g., HTS voice is relatively more intelligible, it requires less training data, it needs very small memory footprint at the time of synthesis, speaker adaptation and prosody modeling are easy. Considering these facts we adapt HTS.

Building HMM-based voice model using HTS toolkit [1] requires a number of steps to follow and involves use of multiple tools. After speech data collection, the primary task is segmentation of speech data at the phoneme level. We use Hybrid Segmentation tool [2] and it requires Festival [6] generated phoneme level and syllable level prompt lab files. Utterance files corresponding to time-aligned phoneme level lab files (generated by Hybrid segmentation tool), Question set for tying phone models, and Speech files in raw file format are provided to HTS toolkit to build HTS voice models. This HTS voice model is put into separate toolkit to synthesize speech waveforms from the given text input. Dealing with these steps are time consuming, tedious and perplexing to a user. Facing these difficulties in HTS voice building process and lack of any document that covers all steps from start to end, motivated us to integrate all our efforts regarding HTS voice building at one place and to develop a toolkit, which can build multiple language HTS voices using a single command. It internally performs all required tasks. This toolkit also provides a framework to synthesize speech waveforms from HTS models. This synthesis framework uses sentence-wise parallel processing to reduce synthesis time for real time uses. This toolkit is available under GPL license at: *https://github.com/TTS-cdac-mumbai/TBT*

2. HMM based TTS building workflow

HMM based TTS voice is built using HTS toolkit and it needs four inputs:

Speech files in raw file format (48 KHz): Speech files are recorded in wav file format at 48 KHz and these are converted into raw file format. Speech parameters are extracted from these raw files.

Context dependent utterance files: Utterance files are prepared using Festvox toolkit, which requires time aligned phonemic transcription as input. Quality of synthesized speech is primarily dependent on accuracy and consistency of time aligned boundary. Creating a properly aligned phonetic transcription is a major challenge of TTS voice building. We use Hybrid Segmentation tool for segmentation, which gives better result, especially for Indian languages in comparison with other available tools. [2] Inputs to this Hybrid Segmentation tool are: syllable level prompt files, phoneme sequence of syllables (present in training data), and phoneme groups (affricates, fricatives, nasals, semivowels, sibilants, fricatives, silence, stops, consonants and vowels). Syllable level prompt files are generated using Festvox [5], which involves Festvox set up, phoneset creation, providing wave files (16 KHz), text file and letter-to-sound rule, etc. Festvox works with 16 KHz speech files, hence, 48 KHz wave files are down sampled to 16 KHz. Output of segmentation is phoneme level time aligned transcription files.

Question Set: Question set is used to create phonetic decision tree in which a yes/no phonetic question is attached to each node. This decision tree is used for state tying and it is the solution to new unseen contexts. Question set needs to be prepared or customized for individual language.

F0 Range: The quality of hts voice is predominantly dependent on the correct minimum and maximum value of F0. Minimum and maximum F0 values are computed from randomly selected wave files from the given data.

Once the hts voice is built, it is used to synthesize test sentences using hts engine, flite hts engine or similar tool. Flite hts engine takes direct text input, but by default it supports English language only and its text processing and text normalization section needs to be customized for new language. To use hts engine, utterance files for test sentences need to be created

Table 1: *Comparison of Synthesis time difference (CPU: Intel core i7-5500U 2.4GHz)*

No. of Sentences	Seq Synth (uSec)	Parallel Synth (uSec)
1 Sentence	0.95	0.95
10 Sentences	10.26	4.31
20 Sentences	20.34	7.52
40 Sentences	41.39	15.46
80 Sentences	85.42	32.13

using Festvox toolkit, and that is passed to it.

Except speech data collection and transcription correction, this toolkit (TBT) does all the tasks mentioned above automatically, and provides a very simple and efficient way to build hts voice using a single command. User has to provide only the wave files and corresponding text files as the input and run the below command.

make GENDER="male/female" LNG="language_name"

In the following sections, we briefly explain the workflow and major features of TBT toolkit.

2.1. F0 range detection

Setting the proper pitch value is important for HTK, otherwise clipping occurs. System determines the minimum and maximum value of F0 for used speech data set. Framewise F0 values for all wave files are calculated using snack tool, and then global minimum and maximum value of F0 is determined. There are chances that at some places value of F0 is exceptionally low or high, to exclude these exceptional values, gender-wise upper and lower threshold range is defined and number of occurrences of F0 value is considered for avoiding the outliers.

2.2. Language Repository

Toolkit includes language repository for individual languages, which contains master files of that language. One can modify or add the language specific rules in these master files. Appropriate contents from the repository are used at the time of voice building as well as during speech synthesis. Currently, TBT contains language repository of thirteen Indian languages viz. Hindi, Marathi, Bengali, Tamil, Telugu, Malayalam, Odia, Gujarathi, Assamese, Manipuri, Kannada, Bodo and Rajasthani. A new language repository can be added easily. A unified parser is developed to handle parsing (letter-to-sound rule, syllabification, phonemefication, etc.) across different languages [4]. New languages can be added by adding custom rules for the corresponding language.

2.3. Parallel processing

In order to reduce the voice building task time, toolkit does parallel processing at the following places:

- Phoneme level and syllable level prompts are generated simultaneously.
- F0 range is calculated in parallel with Hybrid Segmentation Process.
- Raw files generation and phoneme level UTT files generation are done in parallel.

2.4. Regeneration of train.scp

At the beginning of HTS voice model building process train.scp file is generated which contains the name of training data files. During HTS voice model building process some training data is discarded due to various reasons. TBT toolkit identifies the discarded training files and updates train.scp file.

2.5. Synthesis Framework

TBT toolkit contains master Synthesis Framework for each language. After the HTS voice module is built, it is moved to a synthesis framework. This synthesis framework uses sentence-wise parallel processing to reduce the synthesis time, Table1 represents the synthesis time difference between Synthesis Framework without parallel processing and with parallel processing.

2.6. Debugging Mode

TBT builds HTS voice in single command only. However, instead of giving one command, we can run step by step in debug mode to get more control over the process.

3. Conclusion

We have presented an open source toolkit that integrates all required tools and steps related to HTS voice building and synthesis at one place. It allows user to just provide speech data and corresponding text file as input and build HTS voice without any hassle. Toolkit includes Language Resources and Synthesis Framework for thirteen Indian languages. It allows to customize the rules for existing languages and add language resources for a new language easily. We are also providing a user manual of this toolkit, which explains and guides the user. Available speech data [3] at *http://www.iitm.ac.in/donlab/tts/* and this TBT toolkit will enable anyone having even a little computer background to make his own TTS voice.

During the Show and Tell event, we will be explaining the components of the toolkit and show how to use this for building TTS system.

4. Acknowledgements

This work is carried out as a research project "Development of Text to Speech Systems for Indian Languages PhaseII", funded by Ministry of Electronics & Information Technology (MeitY), Govt. of India under TDIL Program. The authors would like to acknowledge each consortium members for their contribution.

5. References

[1] Heiga Zen et al *The HMM-based Speech Synthesis System (HTS) Version 2.0* 6th ISCA Workshop on Speech Synthesis (SSW).2007

[2] S Aswin Shanmugam, Hema Murthy, *A Hybrid Approach to Segmentation of Speech Using Group Delay Processing and HMM Based Embedded Reestimation* In INTERSPEECH 2014.

[3] Arun Baby and Anju Leela Thomas and Nishanthi, N. L. and TTS Consortium *Resources for Indian languages,* CBBLR Community-Based Building of Language Resources, Brno, Tribun EU, Czech Republic, pp.37-43 Sept 2016

[4] Arun Baby, Hema A Murthy et al *A Unified Parser for Developing Indian Language Text to Speech Synthesizers* In TSD, LNCS, Volume 9924, pp.514-521, 2016.

[5] *http://www.festival/festvox.org/*

[6] *http://www.cstr.ed.ac.uk/projects/festival/*