Building of Indian Accent Telugu and English Language TTS Voice Model Using Festival Framework

Chevella Anilkumar and Kancharla Anitha Sheela

September 16, 2021

Building of Indian accent Telugu and English language TTS voice model using festival framework

Chevella Anil Kumar[1],

JNTUH College of Engineering, Hyderabad[1].
Department of ECE, Hyderabad, India[1].
chevellaanilkumar@gmail.com

Kancharla Anitha Sheela[2]

JNTUH College of Engineering, Hyderabad[1].
Department of ECE, Hyderabad, India[2].
kanithasheela@jntuh.ac.in

Abstract- Text-to-speech synthesis is the art of designing talking machines, In this modern age of computers and technology, it plays a vital role in human-machine communication. In addition, it gives the output speech for the given input text of a particular language. Speech plays an important role in the language, speaking style and an efficient way of communication among the people. Speech is the primary thing to express their feelings or emotions to the society and its influence is come to leading position in the human lives. There were a lot of hardwired aspects in the building of English Voices for HTS with the Festvox, particularly in the feature names and questions for the cluster procedures in HTS. The goal of this project was to improve the robustness of the connectivity between HTS and festvox for various databases and languages. We will present Clustergen in this article, which was developed within the widely utilized festival/festvox voice suit. It has the advantage of smoothing the data. Available Indian accent text to speech voice models built by festival framework cannot vocalize/Synthesize text which is included Non standard words. The main objective of this paper is to build a TTS voice models for Indian accent English and Telugu Language to synthesize the given text which is included non standard words using Clustergen. In this Text to Speech synthesis, we are going to use the festival, festvox, Speech Tools, SPTK and Linux 16.04 LTS environment to build synthetic voice models of natural speech. Festival is run time synthesis engine to synthesize the text using built voice models. The voice models are generated by the Statistical Parametric (Clustergen) Speech synthesis technique using festvox, speech tool and SPTK with the help of festival frame work.

Key words: Speech synthesis, TTS, festival, festvox, Speech Tool, SPTK, Clustergen.

1. Introduction

The automatic production of spoken language speech from a written text is commonly referred to as “Text-to-Speech”. And its plays a major role in this advanced technology digital world to speak out the computers. Speech synthesis has multitudinous applications. Some of these include telecommunication services, language education, audio books, interactive voice response (IVR) systems, talking toys, talking ATM’s, vocal monitoring, multimedia with man-machine communication (announcements at public places) and helping the visually challenged people.

TTS systems are broadly divided into three parts. The first part analyzes the form of text in UTF8 or in a transliteration scheme for Indian languages. Secondly, the text is converted into a "linguistic description" and last, a waveform gets generated using this description. which is shown in the below figure 1.

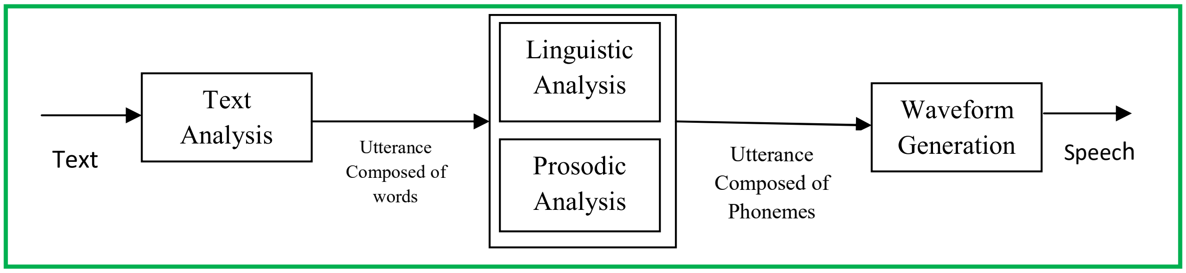


Fig. 1 Text-to-Speech synthesis System

In early days of TTS research [15], Many researchers worked with parametric synthesis, wherein parameters are set by expert-designed formulas. articulatory, formant-based phonemic synthesis, and LPC-based concatenative synthesis are 3 of the most common methods.

The articulatory system attempts to mimic the human articulatory system, including the vocal cords, vocal tract, and so on. Formant synthesis synthesises speech by using formants, which are the vocal tract's resonance frequencies. Although formant synthesised speech is understandable, it is unnatural and robotic in nature. Concatenative speech synthesis creates an utterance by concatenating pre-recorded speech units (phones, syllables). Concatenative synthesis is more natural than the other two methods since it utilises the actual recordings in the database. Concatenative synthesis also has the disadvantage of requiring a huge database. While the aforementioned approaches are still in use. However, new methods have lately been created, and improvements0 have been made to the previous ones. A variety of reasons have contributed to recent development, including

(1) the capacity of computers to execute jobs more quickly,

(2) the proliferation of text and voice databases, and

(3) Advances in voice recognition and synthesis technologies.

This has led to major advancements in voice synthesis. More differences in synthetic speech have resulted as a result of these developments.

2. Literature Review

Despite the fact that text-to-speech encompasses a variety of processes, a great deal of effort has been done to generate speech from speech sounds, instead of going from text to phonemes or even to sound. These are known as just a speech-sound (phoneme) to audio waveform synthesis, instead of going completely from text to phonemes, and now to sound. It was 1936 whenever the U.K. Telephone Company unveiled the first practical use of voice synthesis: a speaking clock. "Noun," "verb," and so on were stored in optical storage and then concatenated to create full sentences. Meanwhile, at Bell Laboratories, Homer Dudley created a mechanical organ-like instrument with pedals & mechanical keys. As long as it was set up correctly, a skilled operator could make it produce noises that sounded as speech. There at 1939 World's Fairs in New York and San Francisco, it was called the Voder.

For analogue electrical systems capable of simulating human speech, it must have been discovered that now the speech signal could've been decomposed into a source-and-filter model with both the glottis serving as a sound source as well as the oral tract working as both a filter. Another example is indeed the vocoder, which was also invented by Homer Dudley. Inside the 1940s and 1950s, most of the effort in synthesis were focused on building copies of both the signal exactly, instead of generating those phones from such an abstract sense like text or speech. Another kind of synthesis that relied on signal breakdown was formant synthesis, in which groups of signals was combined to produce speech that could be recognised. In the past, and even today, this has been challenging to forecast parameters which compactly describe the signal without sacrificing any information essential for reconstructed data. When formant synthesis was first developed, it was possible to specify the formants via hand, with automated modelling as just a goal. Such systems may generate high-quality, identifiable speech when the settings are appropriately tuned. There is still some difficulty in obtaining a completely realistic sounding voice from these devices, even when the process is entirely automated It became easier to concatenate natural recorded speech as digital representations, digital signal processing and inexpensive, general-purpose computer hardware became increasingly prevalent. These two neighbouring half-phones (phoneme realisations that are context-dependent) were brought together to form a single unit. Since a result, mid-phone is a preferable location to concatenate units, as stable places have, by definition, minimal fast change, while borders have sudden changes that rely on the preceding or next unit.

While concatenative synthesis started to gain popularity in the 1970s, large-scale electronic storage have essentially made it feasible. Less resource-intensive methods were worth their weight in saved cycles in gold, to use an unusual metaphor. In spite of the fact that formant synthesis uses less storage, still it needs a lot of processing power. Voice compression techniques were created to make it easier to utilise speech within applications. Early forms of mass-produced voice synthesisers include the Texas Instruments Speak 'n

Spell toy introduced in the late 1970s. To our contemporary eyes, the film was of low quality, but in terms of its period, it was remarkable. Most of the words and letters used during speech were separate, and there were a few phrases created by concatenation. Speech was encoded utilizing LPC (linear Predictive Coding). Home computers like the BBC Micro inside the UK as well as the Apple became popular with simple text-to-speech (TTS) engines built on specialised microchips. Mitt's MITalk synthesiser, designed by Dennis Klatt, changed the way people thought about automated speech synthesis for the rest of the globe. As DECTalk, it generates slightly robotic, but highly intelligible, speech that may be heard and understood by anybody. Formant synthesiser, representing the state-of-the-art in its day, it was released in 1976.

As early as the early 1980s, speech synthesis investigation was only conducted in big labs that had the resources to do so. By the mid-80s, even as cost of something like the hardware fell, more laboratories and colleges began to participate. To some extent, software-based synthesisers were available by the late 1980s, but the quality of both the voice was still unmistakably human and could be produced in near real-time. Naturally, with faster computers and greater disc space, individuals started to seek for ways to improve synthesis by utilising larger, and more diverse stocks for concatenative speech synthesis. Nuu-talk was created in the late 1980s and early 1990s by Yoshinori Sagisaka of Advanced Telecommunications Research (ATR) in Japan. But instead of 1 example of every diphone unit, there might be several. A distance system assists on acoustics was used to identify the optimum selection of sub-word units from an extensive collection of common speech. A considerably simpler phonetic structure than English made it feasible to get high-quality results with relatively small datasets in Japanese. Also in 1994, it took many days of CPU time to produce the parameter files for a new nuu-talk voice (503 senetences), and synthesis really wasn't usually feasible in real time.

Unit selection has become a hot issue in speech synthesis research when Rob Donovan's PhD work demonstrated general unit selection synthesis in English, and ATR's CHATR system at the end of the 1990s. Despite some examples of it operating well, generalised unit selection is renowned for occasionally delivering poor quality synthesis. Because the best search and selection algorithms aren't 100 percent dependable, both high and low quality synthesis is created, and many challenges remain in turning generic corpora into high-quality synthesisers as of this writing.

a new statistical technique of voice synthesis has gained prominence in the new millennium (2000). Professor Tokuda's HTS System (from Nagoya Institute of Technology) demonstrated that constructing generative models about speech, instead of choosing unit examples, may produce dependable, high-quality voice. As a result of the Blizzard Challenge in 2005, HTS output was clearly comprehended by listeners. If indeed the data becomes less reliably recorded, both HTS and so-called HMM synthesis appear to perform better than unit selection, which appears to need extremely large properly labelled corpora. However, HTS tends to receive from either the Festival voice creation toolset, which is closely linked with the CMU CLUSTERGEN statistical parametric synthesiser.

3. Methodology

3.1 Architecture for Text to Speech

A text-to-speech synthesis mechanism has been implemented in Festival. There are three main components to the TTS process that [7] can identify. ***Text analysis:***Identified words and basic utterances from a raw text. ***Linguistic and Prosodic analysis****:* Going to assign prosodic structure for words, including phrasing, intonation, and durations. ***Waveform generation:*** Produce a waveform from the a fully defined form (pronunciation and prosody). Typical Text-to-Speech (TTS) architecture is shown in figure 2.

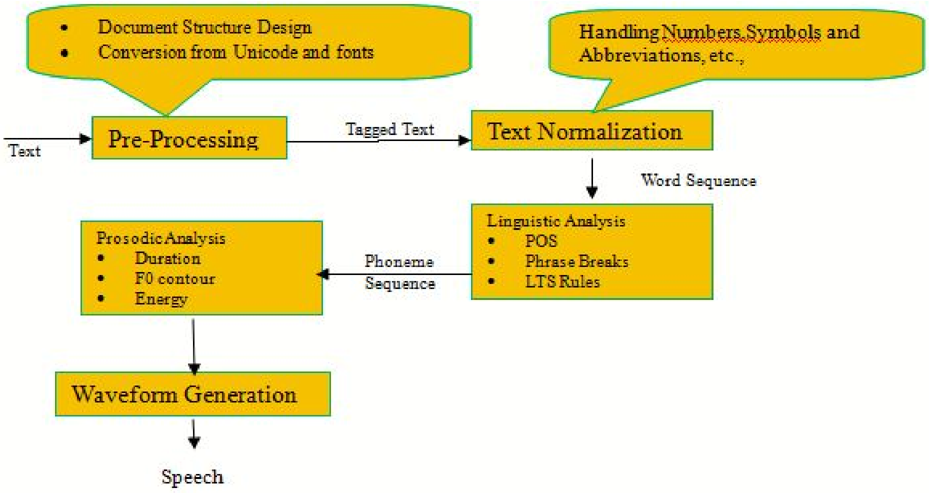


Fig. 2 Architecture of Text to Speech

3.2 Text analysis

Identifying words in a document is called text analysis. For the purposes of this project, words refer to tokens since there is a well-defined way of determining their pronunciation from a lexicon and utilising the laws of letter to sound correspondences. The initial step in text analysis would be to tokenize the input text. In Festival, we also fragment the text at about this point into more manageable utterances. Information for what could be characterised as such a sentence is stored in such an utterance structure.

Text from electronic documents, news articles, blogs, emails, etc. is perhaps the most common input for a text to voice system in the real world. There is no such thing as a conventional dictionary when it comes to real-world text. Numbers, abbreviations, homographs and punctuation characters including such exclamation '!', smileys ':-)', etc., are used in the text. As a result of the supplied text, the module attempts to normalise non-standard words, anticipate prosodic pauses, and create the proper phone sequences with each of the words it encounters.

Normalization of Non-Standard words

Text normalisation is another function of text analysis, in addition to chunking. Most of the time, tokens used in text don't really correspond to their pronunciation. The much more apparent example is numbers. Observe the following phrase.

On April 3 1992, the JNTUH University bought 1992 computers.

Tokens consisting only of numbers may be pronounced in a variety of ways in English. Since it is the 3rd day of the month, this "3" above has been pronounced "third," and an ordinal, the very first "1992" being pronounced "nineteen ninety two," as well as the second "1992" is sounded "one thousand nine hundred and ninety two," as that is a number.

There are two issues that arise here: non-trivial token-word relationships and homographs, in which the same token can have various pronunciations within various contexts. Homograph disambiguation was considered part of text analysis in Festival. Aside from numbers, there seem to be a variety of additional symbols with structural properties that need specific processing, like money, times, and addresses. Token-to-word semantics may be used to deal with many in Festival. These really are unique to the English language (and sometimes text mode specific).

3.3 Linguistic and Prosodic Analysis

This would be the second step in text-to-speech conversion. We now also have our words; all we need to do now is come up with something else to say. We will need segments (phonemes), durations for them, as well as a format frequency (F0) for this. Although this may seem to be naïve way, the quality of the speech, how natural, intelligible, and acceptable it really is, is mainly determined by the appropriateness of both the phonetic & prosodic output. Languages prosody is utilised to provide variety for emphasis, contrast, and general ease of comprehending and conveying the information towards the listener. This degree of diversity in prosody is heavily influenced by the language being spoken. Many things were lexical, and they cannot be altered without altering the words that are uttered. In English, for example, lexical stress includes part of both the definition of words; changing the location of the emphasis may alter the word (e.g., from "pro'ject" (noun) to "proje'ct" (verb)). Because poor prosody makes statements difficult to comprehend, it is critical for a voice synthesis system should produce proper prosody.

3.4 Waveform generation

This will be the last and most crucial component of the festival speech synthesis system[1]. Using the festvox [3][4] tool, this step gets phone metadata, prosody for synthesis from of the previous block, and existing voice models. It will generate synthetic speech as just an output by mixing all of these factors. The waveform synthesiser differs depending mostly on voice models so order to obtain the necessary and needed information from of the voice models and create synthetic speech. This table below depicts the Synthetic flow of the festival Speech Synthesis system..

Table 2: Synthetic flow of festival Speech Synthesis

Pipeline Stage	Example	Function	Description
Text	June 25	Textify	ASCII or utf-8 character string Text to token sequence conversion Word-like components that can be processed
Tokens	June, 25	Token_ Token_POS	Convert tokens to words
Words	June twenty fifth Noun Noun Noun	POS Phrasify Word Pauses	Lexical elements Classify a word’s part of speech Split utterance into phrases Convert words to sound segments Insert pauses at phrase breaks
Phonemes	Jhuu n t w e n t ii f i f th	Duration PostLex Intonation	Pronounceable elements Predict segment durations Generate F0 interpolation points
Generation	---	WaveSynth	Synthesis method used to generate the speech signal (di-phone, unitsel, HTS, Clustergen)
Waveform	---		Final result

Statistical Parametric Synthesis (SPS)

One of the most recent developments in TTS is Statistical Parametric Synthesis (SPS) [23]. The SPS techniques generate speech by learning a set of criteria from the speech samples. Unlike conventional parametric synthesis techniques, which involve human parameter definition and hand-tuning, SPS methods estimate the parameters of speech sounds and their dynamics using statistical machine learning models such as CART, HMMs, and others. SPS techniques simplify storage through encoding speech data in terms of a small number of parameters, and they also give mechanisms for manipulating prosody, voice conversion, and so on. When compared to natural and often inconsistent speech produced by unit selection approaches, SPS methods generate understandable and consistent speech. Statistical Parametric Synthesis has sometimes been called "HMM-generation synthesis",[13] to distinguish it from HMM-state sized units in unit selection, however in this work there is no actual requirement for HMMs. No HMMs are used at synthesis time, even though HMMs can be used to label the data. Therefore, Statistical Parametric Synthesis seems a better method. It has the advantage of **smoothing** the data. There are two different types of Statistical Parametric speech synthesis Techniques:

- HMM-Based Speech Synthesis (HTS)
- Clustergen Speech Synthesis

In this project we will present CLUSTERGEN [19], a Statistical Parametric Synthesizer that has been created within the widely used Festival/FestVox [1][3] voice building suite. the basic CLUSTERGEN system is only slightly different from HTS.

HMM Based Speech Synthesis

Even though many speech synthesis systems could produce high-quality speech, they cannot produce speech with just a variety of voice characteristics including such speaker individuality, speaking styles, moods, and so on. A significant quantity of speech data is required to acquire different voice qualities in speech synthesis systems regarding the selection and concatenation of acoustical units. Nevertheless, collecting and storing such voice data is challenging. This HMM based speech synthesis system, HTS [13] is developed in order to build speech systems that can produce a variety of voice characteristics.

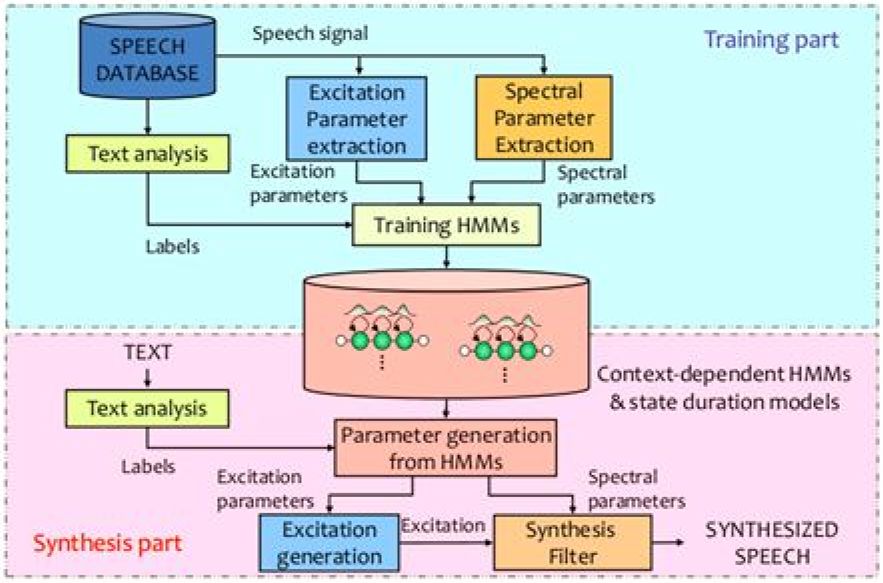


Fig. 9 HMM Speech Synthesis

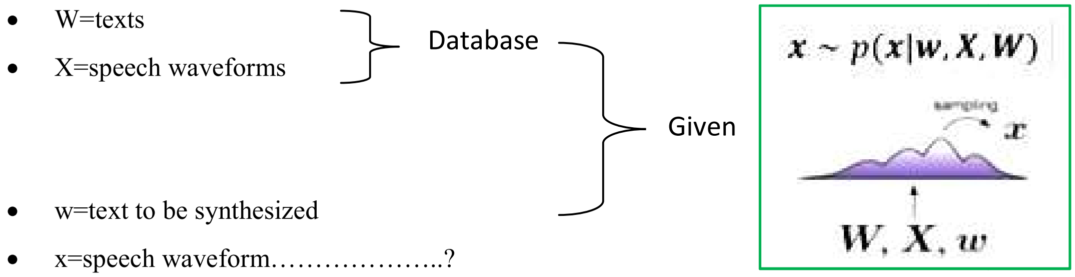
The diagram above depicts a high-level overview of the HMM speech synthesis system. As indicated with in image, two phases are involved with HMM-based speech synthesis: training and synthesis.

The training section's purpose is to conduct statistical modelling of speech based on features collected from the speech database and corresponding transcription text. The retrieved properties include not just prosodic and vocal tract information, but also contextual and durational models. This HMM-based voice synthesis framework models pitch and spectrum simultaneously. Prosody is expressed as logF0 in spectral representation and Mel-based cepstral coefficients in prosody representation. To address the issue of non-continuous pitch values within unvoiced areas, Multi Space Probability Distribution modelling is used. Furthermore, context clustering is done using decision trees in order to fully utilize contextual information there at lexical and syntactic levels; duration models are also built concurrently.

During the synthesis stage, an arbitrarily chosen text to be synthesized is first transformed to a context-based label sequence. Second, a sentence HMM is built using the label sequence through concatenating context dependent HMMs. The phrase HMM's state durations have been determined in order to maximize the output probability of state durations, then a sequence of mel-cepstral coefficients and values, including voiced / unvoiced decisions, has been determined in order to maximize the output probability of the HMM and used the speech parameter generation algorithm. This usage of dynamic features is the system's primary characteristic: by include dynamic coefficients inside the feature vector, every speech parameter sequence produced in synthesis was bound to be realistic, as specified either by statistical parameters of both the HMMs. Finally, the MLSA filter is being used to directly synthesize the speech waveform from the generated mel-cepstral coefficients and values since the source-filter model of speech production as well as reconstruction employs multiple kinds of excitation signals, allowing it to produce buzzy and high quality speech as output.

4. Statistical formulation of Speech Synthesis

This would be the fundamental issue of speech processing; we have a speech database and a set of pair of texts and corresponding speech waveforms. Given a text to be synthesized and what is the speech waveform corresponding to the text?



- W=texts
- X=speech waveforms

Database

Given

- w=text to be synthesized
- x=speech waveform………………..?

Estimating Predictive Distributionis hard.

Introduce generative representation (λ): model parameters

$$p(x|w,X,W) = \int p(x|w,\lambda)p(X,W)d\lambda$$

lambda is the motor parameters then approximating integral by maximizing this term P(λ|X,W). you can separate it into the training part and generation part as we know modeling speech waveform.

$$\tilde{\lambda} = \arg\ max\, p(\tilde{\lambda}\,|X,W) - Training$$

$$x{\sim}p(x|w,\tilde{\lambda}) - Waveform\ generation$$

Usually, the generative model is decomposed into sub-modules. Finally, Decompose the generative model into sub-modules:

$$\{\tilde{\lambda}a,\tilde{\lambda}l\} = \arg\ max \int \textstyle\sum_l p(X|O)p(O|L,\lambda a)p(L|W,\lambda l)dOp(\lambda a)p\lambda l) \text{ --Training}$$

$$x{\sim} = \int \textstyle\sum_l p(x|o)p(o|l,\tilde{\lambda}a)p(l|w,\tilde{\lambda}l)do \text{ --Generation}$$

And it is difficult to perform integral and sum then approximated it by step-by-step maximization as shown in below equations and figure.

$$\tilde{\lambda}l: pre-trained\ text\ analysis\ module\ parametr$$

$$\tilde{O} = \arg\ max\, p(X|O)\text{—}\ Speech\ feature\ parameter\ extraction$$

$$\tilde{L} = \arg\ max\, P(L|W,\tilde{\lambda}l)\ or\ P(\tilde{O}|L,\tilde{\lambda}a)\ or\ P(\tilde{O}|L,\tilde{\lambda}a)P(L|W,\tilde{\lambda}l)\text{—}\ labeling$$

$$\tilde{\lambda}a = \arg\ max\, P(\tilde{O}|\tilde{L},\lambda a)\, p(\lambda_a)\text{—}\ acoustic\ model\ training$$

$$\tilde{l} = \arg\ max\, P(l|w,\tilde{\lambda}l)\text{—}\ Text\ Analysis$$

$$\tilde{o} = \arg\ max\, p(o|\tilde{l},\tilde{\lambda}a)\text{—}\ Speech\ parameter\ generation$$

$$x \sim p(x|\tilde{o}) - Waveform\ generation$$

4.1 CLUSTERGEN Statistical Parametric Synthesizer

CLUSTERGEN synthesizer [19] is indeed a technique for training models and utilizing these models at synthesis time in the Festival Speech Synthesis System's. The training requires the best databases are those that are phonetically balanced recorded utterances, and corresponding text transcription. The process to synthesize speech is broadly classified into two phases: (1) Training phase, (2) Synthesis phase

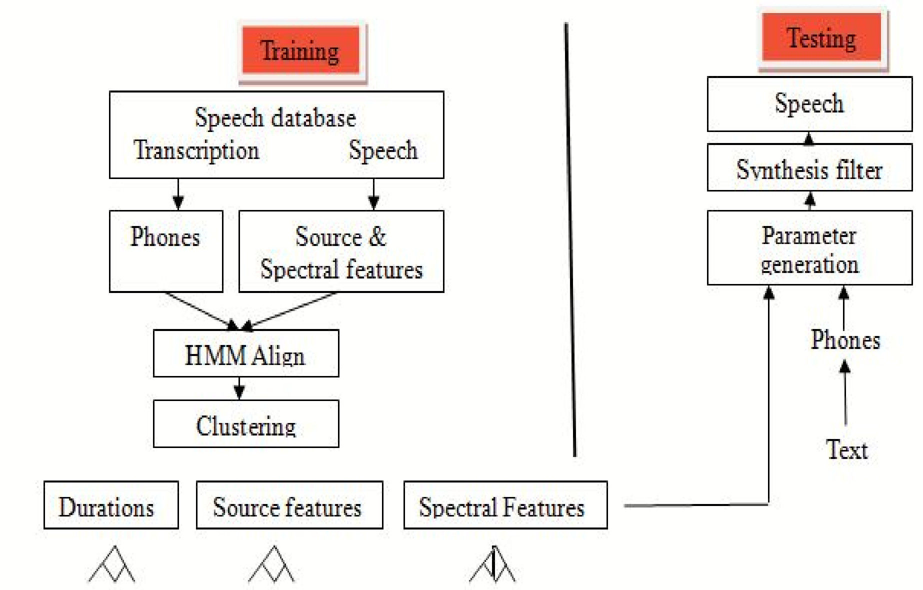


Fig. 10 Block diagram of Clustergen speech synthesis

Training Phase:

In Training phase, label the database automatically using an EHMM labeler which is included in the Festvox. With Baum Welch algorithm, Context independent HMMs are trained to force-align the phonemes derived from transcriptions and audio.

A source-filter model of speech is being used to extract characteristics from a speech signal. There are filter parameters that can be retrieved from the MCEP data, and fundamental frequency estimations may be obtained for each five milliseconds and also for every 5ms, 25 MCEPs are extracted from a speech signal. Across use of the phoneme labels, the F0 value was interpolated by unvoiced areas. As a result, all 5ms frames with voiced or unvoiced speech have a non-zero F0. F0 modelling methods are used here [26]. Five millisecond intervals are used to combine 25 MCEPs with the F0 to produce a 26-feature vector. It is possible to extract high-level characteristics for all of these vectors, such as syllable structure, word location and phonetic properties. unit selection synthesizer uses a similar set of retrieved features, except here, features can be extracted for every vector instead of for every segment or phoneme.

CART Tree builder wagon is used in speech tool for clustering the utterances. This has been expanded to include vector predictees as part of its capabilities. With a waggon, CART trees are constructed to discover questions that divide the data in a manner that minimizes the amount of impurity. The impurity is calculated as

$$N * (\sum_{i=1}^{24} \sigma_i$$

N denotes the number of samples throughout the cluster. and σ_i is the standard deviation for MCEP feature i

Over all samples in the cluster, the factor N helps keep clusters large near the top of the tree thus giving more generalization over the unseen data. Initial studies built joint F0/MCEP models, but slightly better results are obtained when separate F0 and MCEP models are built. To forecast the durations of each HMM state, an extra CART tree is created. Each state duration in clustergen is predicted independently, even though they do include features to identify the states position in its phoneme.

Synthesis Phase:

The phone string is formed from the text in the synthesis phase, as it is in previous Festival synthesis systems, and then an HMM state name connection is built tying each phone to its three sub phonetic elements. The length of each HMM state is predicted using the duration CART tree. To fill the length of the projected state time, a set of empty vectors is constructed. The questions are answered using the CART tree unique to the state name, and the means from the vector at the selected leaf are added as values to each vector. When each state is predicted by a single vector. In CLUSTERGEN[19] predicts several vectors for every state. As a result, the projected vector might well be different depending on the condition in which it is predicted. Each track containing coefficients is smoothed using a basic three-point moving average.

$$\overline{s_t} = \frac{s_{t-1} + s_t + s_{t+1}}{3.0}$$

Where s_t is the sample at time point t. On the other hand, an MLSA filter is used to rebuild the speech from the predicted parameters. Voicing decisions are currently done by phonetic type directly from the labels, rather than trained from the acoustics.

5. Measurement and Testing.

For the purpose of evaluating the quality of synthesized Speech, we utilize Mel Cepstral Distortion (MCD). A waveform often is broken down into a series of multi-dimensional coefficients (vectors) at regular intervals called frames with speech processing systems. Cepstral coefficients of 25-D mel frequency scale with a frame step size of 5 ms is commonly used for TTS applications. It is possible to express this series of frames as Vd(t), where d is the dimension index (ranging from 1 to 24) and t seems to be the time index. The MCD value calculated using the following equation with the two waveforms, target $V^{\mathbf{targ}}$ and reference $V^{\mathbf{ref}}$ having a mean mel ceptral distortion.

$$MCD(v^{targ}, v^{ref}) = \frac{\alpha}{T}\sum_{t=0}^{T-1}\sqrt{\sum_{d=s}^{D}(v_d^{targ}(t) - v_d^{ref}(t))^2}$$

$$\alpha = \frac{10\sqrt{2}}{\ln 10} = 6.14185$$

where the scaling factor α was extant for historical explanations and s is the “starting” dimension of the inner sum, and equals either 0 or 1.

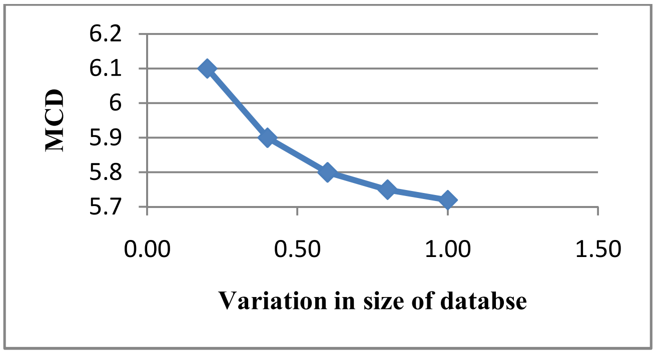
Therefore, the objective measure in our project is given as below equation

$$\frac{10}{\ln(10)} * \sqrt{2\sum_{j=1}^{24}(mc_j^t - mc_j^p)^2}$$

Where mc_j is the j^{th} MFCC coefficient in a frame, $\mathrm{mc}^{\mathbf{t}}$ is the target MFCC we are comparing against and $\mathrm{mc}^{\mathbf{p}}$ is the predicted MFCC. The accepted MCD value should be in the range of 4.0-8.0[19].

And the amount of data in the training data set affects the synthesis results. It is notable that even modest amounts of data can produce quite acceptable synthesis. As the size of the database increased to train the voice model the quality of synthesized speech that the value of Mel Cepstral Distortion (MCD) going to be increased. The follow diagram shows the relation between MCD value and size of the database.

Size of the database to MCD



Known limitations of festival Tool to build the Indian languages TTS Engine:

- To Synthesize the Particular text in TTS from festival environment using CMU Dictionary[2] , The Text should be in double inverted commas, the syntax is shown below,

  (SayText “welcome to research and training unit for navigational electronics”)

  Hence, it cannot be synthesized if the inverted commas are in the text itself. So, before calling the function text2wav, invited commas in the text should be removed. the following command can be used to remove the quotes from a text document

  sed ‘s/” / /g’ input_file.txt > output_file.txt # is inserted To remove the inverted commas from the text file before the following command.
- The digits with a decimal point and Some of the Non-Standard words not getting vocalized like Dr, Prof., etc) for indic languages, we need to define the lexicon rules manually.

Training data set /audio file used

- For TTS, we used Studio Recorded Indian accent English speech dataset [9][10][20] collected by IIT Madras Speech and Music Technology lab and also recorded at JNTU Hyderebad. The given speech data is segmented into .wav files with length of 5sec-15sec and prepared corresponding text Prompt file for training the database.

- Each .wav file consists of 16 kHz sampling frequency and quantized with signed 16bits/sample with mono stream.

6. Results and Conclusion

Cmu_indic_aniljntuhenglish_cg, Indian accent English voice model generated using festival Clustergen speech synthesis system With training speech data of 4 hours 10 minutes, 2 hours, and 1 hour, we obtained MCD values of 5.24, 5.31, and 5.42, respectively, and the output of TTS system played with Signed 16 bit, Rate=16000 Hz, with mono stream, and the distributed voice model is available as **festvox_cmu_indic_aniljntuhenglish_cg.tar.gz**.

Cmu_indic_aniljntuhtelugu_cg, an Indian accent The Telugu voice model was generated using the Festival Clustergen speech synthesis system with training speech data of 5 hours, 2.5 hours, and 1 hour, yielding MCD values of 5.15, 5.28, and 5.38, respectively, and the output of the TTS system was played with Signed 16 bit, Rate=16000 Hz, with mono stream, and the distributed voice model. The voice model is available as **festvox_cmu_indic_aniljntuhtelugu_cg.tar.gz.**

7. Conclusion and future scope

Clustergen is a generative-based speech synthesis method. It is a new type of synthesis, and there is still much work to be done to make the synthesised speech sound as natural as human voice. We used the festival tool to create the Indian accent English and Telugu TTS Engine, and the output of Clustergen speech synthesis produces a distortion-like buzz. As a result, there is still much to be done to improve it. We can also extend this work to sentiment analysis of synthesised speech for classifying robot emotions, and so on.

8. References

1. A.W. Black, P. Taylor, and R. Caley, “The Festival speech synthesis system,” http://festvox.org/festival/, 1998.
2. CMU, “The carnegiemellon university pronunciation dictionary,” www. speech.cs.cmu.edu/cgi-bin/cmudict, 2008.
3. http://festvox.org
4. http://festvox.org/examples/cmu_us_kal_diphone/
5. ftp://ftp.cstr.ed.ac.uk/pub/festival/
6. http://www.festvox.org/festival/downloads.html
7. http://www.festvox.org/docs/manual-1.4.2/festival_toc.html
8. http://www.festvox.org/docs/speech_tools-1.2.0/book1.htm
9. Arun Baby, Anju Leela Thomas, et.al. “Resources for Indian languages”
10. IIT Madras. Indic tts - voices. https://www.iitm.ac.in/donlab/tts/voices. php.
11. Thierry Dutoit “An introduction to text-to-speech synthesis” Springer Science & Business Media, volume 3, 1997.
12. H. Zen, K. Tokuda, and A. Black, “Statistical parametric speech synthesis,” Speech Commun., vol. 51, no. 11, pp. 1039–1064, 2009.
13. K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, “Speech parameter generation algorithms for HMM-based speech synthesis,” in Proc. ICASSP, 2000, pp. 1315–1318.
14. H. Zen, T. Toda, M. Nakamura, and T. Tokuda, “Details of the Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005,” IEICE Trans. Inf. Syst., vol. E90-D, no. 1, pp. 325–333, 2007.
15. Klatt DH. Review of text-to-speech conversion for English. J AcoustSoc Am. 1987 Sep;82(3):737-93. doi: 10.1121/1.395275. PMID: 2958525.
16. R. Carlson, B. Granström “A text-to-speech system based entirely on rules” DOI:10.1109/ICASSP.1976.1169952.
17. Andrew J. Hunt and Alan W. Black “unit selection in a concatenative speech synthesis system using a large speech database” 0-7803-3 192-3/96 $5.0001996 IEEE.

18. Tim White , Alan W Black, et.al. “Open-Source Consumer-Grade Indic Text To Speech”, 9th ISCA Speech Synthesis Workshop, September 13 – 15, 2016, Sunnyvale, CA, USA.
19. Alan W Black “CLUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling”, INTERSPEECH 2006 – ICSLP.
20. S. P. Panda, A. K. Nayak, and S. Patnaik, “Text to Speech Synthesis with an Indian Language Perspective”, International Journal of Grid and Utility Computing, Inderscience, Vol. 6, No. 3/4, pp. 170-178, 2015.
21. John Kominek “TTS From Zero Building Synthetic Voices for New Languages”, Ph.D thesis CMU-LTI-09-006.
22. Kishore Prahallad “Automatic Building of Synthetic Voices from Audio Books” Ph.D thesis CMU-LTI-10-XXX July 26, 2010.
23. Heiga Zen a,b,*, Keiichi Tokuda a , Alan W. Black c “Statistical parametric speech synthesis” Speech Communication 51 (2009) 1039–1064.
24. ElokCahyaningtyas, DhanyArifianto “Synthesized Speech Quality of Indonesian Natural Text-to-Speech by using HTS and CLUSTERGEN” 2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Technique (O-COCOSDA) 26-28 October 2016, Bali, Indonesia.
25. VijayadityaPeddinti “Synthesis of missing units in a Telugu text-to-speech system” master of science thesis, July, 2011.
26. Prahallad, Kishore & Black, A.W. &Mosur, R.. (2006). Sub-Phonetic Modeling For Capturing Pronunciation Variations For Conversational Speech Synthesis. 1. I - I. 10.1109/ICASSP.2006.1660155.