PUNJABI SPEECH SYNTHESIS SYSTEM USING HTK

Divya Bansal[1], Ankita Goel[2], Khushneet Jindal[3]

School of Mathematics and Computer Applications,
Thapar University, Patiala (Punjab) – India
[1]divyabansal150@yahoo.com
[2]goel.ankitathapar@gmail.com
[3]khushneet.jindal@thapar.edu

ABSTRACT

This paper describes an Hidden Markov Model-based Punjabi text-to-speech synthesis system (HTS), in which speech waveform is generated from Hidden Markov Models themselves, and applies it to Punjabi speech synthesis using the general speech synthesis architecture of HTK (HMM Tool Kit). This Hidden Markov Model based TTS can be used in mobile phones for stored phone directory or messages. Text messages and caller's identity in English language are mapped to tokens in Punjabi language which are further concatenated to form speech with certain rules and procedures.

To build the synthesizer we recorded the speech database and phonetically segmented it, thus first extracting context-independent monophones and then context-dependent triphones. For e.g. for word bharat monophones are a, bh, t etc. & triphones are bh-a+r. These speech utterances and their phone level transcriptions (monophones and triphones) are the inputs to the speech synthesis system. System outputs the sequence of phonemes after resolving various ambiguities regarding selection of phonemes using word network files e.g. for the word Tapas the output phoneme sequence is ਤ,ਪ,ਸ instead of phoneme sequence ਟ,ਪ,ਸ.

KEYWORDS

Hidden Markov models, Context-dependent acoustic modeling, Punjabi speech corpora.

1. INTRODUCTION

Speech is the most important form of communication in everyday life. However, the dependence of human computer interaction on written text and images makes the use of computers impossible for visually and physically impaired and illiterate masses [1]. Text-to-speech synthesis (TTS) helps speech processing researchers to act upon this problem by synthesizing speech (in local languages e.g. Tamil, Hindi, Punjabi etc.) from written text like in browsers, mobile phones etc.
Speech can be synthesized by mainly three methods: Articulatory synthesis, Concatenative synthesis and Formant synthesis

Articulatory synthesis tries to model the human speech production system (especially vocal tract system, various articulators viz. lip, tongue, jaw etc.) and articulatory processes directly. However, it is also the most difficult method to implement due to lack of knowledge of the complex human articulation organs.

Concatenative speech synthesis systems can synthesize high quality and more natural sound speech but in order to synthesize speech with various voice characteristics such as speaker individualities, speaking styles, emotions, etc., a large amount of speech corpus and memory is required as stored basic speech units (like syllables, diphones etc.) are concatenated to form word sequence using pronunciation dictionary.

Formant synthesis is based on the rules which describe the resonant frequencies of the vocal tract. The formant method uses the source-filter model of speech production, where speech is modeled by parameters of the filter model [2]. Rule-based formant synthesis can produce quality speech which sounds unnatural, since it is difficult to estimate the vocal tract model and source parameters [3].

One more approach for speech synthesis is **Hidden Markov Model based synthesis** i.e. HTS. It was initially implemented for Japanese language but, today, can be implemented for various languages viz. Hindi, English, Tamil etc. It is used easily for implementing prosody and various voice characteristics on the basis of probabilities without having large databases. In this approach speech utterances are used to extract spectral (Mel-Cepstral Coeff.), excitation parameters and model context dependent phone models which are, in turn, concatenated and used to synthesize speech waveform corresponding to the text input.

This paper is organized as follows: In section 2 we present Hidden Markov Model based speech synthesis, section 3 describes overall implementation of Hidden Markov Model based Text-to-Speech System on Hidden Markov Model Toolkit architecture from feature extraction to training of system, the fourth part contains results of speech synthesis and finally the fifth part concludes the paper with Discussion and Conclusion.

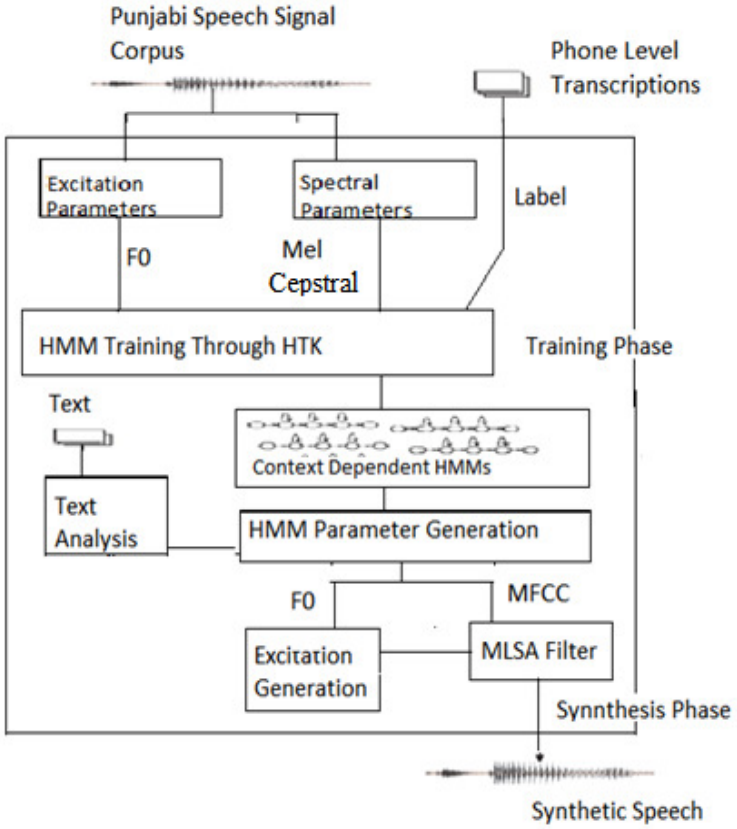


Figure 1. HMM Based Speech Synthesis System

2. HIDDEN MARKOV MODEL BASED SPEECH SYNTHESIS

In speech synthesis, Viterbi algorithm is used to find the most probable path through Hidden Markov Models that can generate speech signal feature vectors like MFCC (Mel Cepstral Coeff.) which are used, in turn, to generate speech signal.

2.1 Training Part

In this the spectral parameters i.e. Mel Cepstral Coefficients and excitation parameters i.e. fundamental frequency *F*0 are extracted from the speech database and concatenated further to use them for Hidden Markov Models training acoustic models. The training of phone Hidden Markov Models using pitch and Mel cepstrum simultaneously is enabled in a unified framework by using multi-space probability distribution Hidden Markov Models and multi-dimensional Gaussian distributions [4]. The simultaneous modeling of pitch and spectrum results in the set of context-dependent Hidden Markov Models. [2]

2.2 Synthesis Part

In this part, the speech parameters like Mel Frequency Cepstral Coefficients etc. are generated according to the text given as input from the context dependent Hidden Markov Model phone models e.g. for word tapas phone model is t-a + p etc. which are obtained as output from the

training part. These generated speech parameters are, in turn, used to synthesize speech signal as final output.

This approach is very flexible as is implemented by the acoustical features of phone models obtained from speech corpora. Thus characteristics of synthesized speech can easily be modified by altering Hidden Markov Model parameters and acoustical features.

3. HTS IMPLEMENTATION ON HTK ARCHITECTURE

3.1 Signal Features Generation

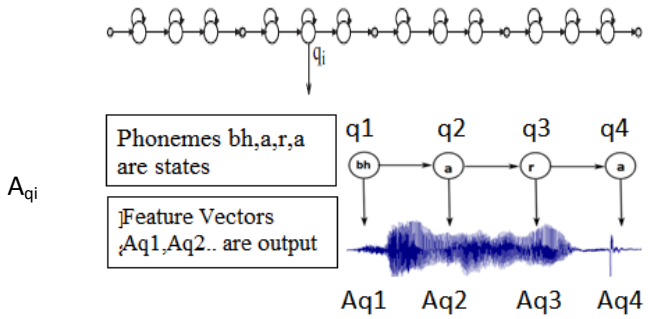
HMM (Hidden Markov Model) have three model parameters (A, B, π) that is there are finite number, say N, of states in Hidden Markov Model. At each time t, a new state is entered based on the transition probability distribution (A) which depends on previous state. After each transition, an observation output symbol depends on the current state based on output probability distribution (B) and π is initial state probability.

In order to synthesize speech, most probable sequence of state feature vectors $\hat{A}$ is required to find from Hidden Markov Model λ, which contains concatenated context-dependent triphones like t-o+n or context-independent monophones like t, o, n (phone transcriptions) corresponding to the symbols in a word w like Tony which is present in text that is required to be synthesized. These acoustic phone models are obtained in training phase after modeling Hidden Markov Models by various feature parameters obtained from stored speech corpora.

Thus we need to generate feature vector sequence $\hat{A} = A_{q1}, A_{q2}, A_{q3} \ldots A_{qL}$ of length L by maximizing the likelihood P(A | λ) of a Hidden Markov Models

$$\hat{A} = \arg\max \{P(A \mid \lambda)\}$$
$$= \arg\max \{\sum_Q P(A \mid q, \lambda)\, P(q \mid \lambda)\} \quad (1)$$

In this equation P(A | λ) of a Hidden Markov Model is calculated by adding the product of joint output probability P(A | q, λ) and state sequence probability P(q | λ) over all possible paths Q [4] Where Q = q1, q2., qL is the path through the states of the model λ and q_i is a state at time t_i as in Fig. 2



2(a) 2(b)

Figure 2(a). Concatenated HMM chain 2(b). HMM chain for word bharat

Thus we are using Viterbi approximation as we need to find most probable state sequence for generating feature vector sequence Â because searching for all possible paths through the model is time consuming and complex.

The state sequence $\hat{q}$ of the model λ can be maximized independently of Â

$$\hat{q} = \arg\max \{P(q \mid \lambda, L)\} \quad (2)$$

Hidden Markov Model Toolkit represents output distributions by Gaussian mixture densities. Thus output probability distribution of each state *qi* is represented by one Gaussian density function with a mean vector μ_i and covariance matrix $\sum_i$. The Hidden Markov Model λ is a set of all means and covariance matrices for all N states:

$$\lambda = (\mu_1, \sum_1, \mu_2, \sum_2, \mu_3, \sum_3, \mu_N, \sum_N). \quad (3)$$

During Hidden Markov Model modeling of acoustic models, means vector μ_i and covariance matrix $\sum_i$ are calculated initially from features extracted from speech corpora and re-estimated for each state of all phone models.

3.2 Data Preparation and Feature Extraction

The training of Hidden Markov Model models and testing of speech synthesis system require speech utterances and their phone level transcriptions. Punjabi speech corpora used for training the system contains speech utterances of one female speaker. In phase-I we have considered recording data that consists of 61 words (i.e words starting with letter ਤ and ਟ) that are arranged in 17 samples and in phase-II training data of 81 words (words containing ਅ and ਆ) arranged in 23 samples is considered. The data is recorded using microphones at room environment. Distance between speaker's mouth and microphone is approximately 5-7 cm.[8] Samples are recorded at a sampling rate of 8000 Hz, 16 bits bit depth and mono channel using Power Sound Editor. Recorded speech files are stored in .wav format.

For training Hidden Markov Models each recorded sample need to have corresponding phone level transcriptions. This is done using Hidden Markov Model Toolkit label editor HLEd that generate phone level MLF (Master Label File) by using mkphones.led edit script. E.g. For sample word "Tony" phone transcription generated is

```
# !MLF! #
"*/S0001.lab"
sil t o n i
```

Figure 3. Phone Transcription File (Phones0.mlf)

Further, the recorded speech samples are parameterized into sequence of excitation and spectral features. For this Mel Frequency Cepstral Coefficients (MFCCs), which are derived from FFT

(Fast Fourier Transform) based log spectra are used. All input .wav files are converted to Mel Frequency Cepstral Coefficient vectors by using HCOPY tool of Hidden Markov Model Toolkit. The speech signals were windowed using a 25 ms Blackman window and 10 ms frame period. The spectral feature vector consisted of 39 mel-cepstral coefficients including the zeroth coefficient (=13) and its delta coefficients (=13) and acceleration coefficients (=13).

3.3 Training of Hidden Markov Model

Initially for training of Hidden Markov Model a prototype model, proto, is defined. It is initialized using HMM Toolkit tool HCompV that computes the global mean and variance and set all of the Gaussians in a given Hidden Markov Model to have the same mean and variance. 5-state left-to-right Hidden Markov Models with no skips are used in which first and last states are non-emitting states. The system is trained for 27 monophones models. These flat start monophones stored in various hidden markov models directories are re-estimated using the embedded re-estimation tool HERest following Baum-Welch Re-estimation theoretically. For each state of all the monophones mean and variance vectors are estimated. After that the triphones models were made out of monophones models and trained using HERest. These triphones are created from monophones by HLEd tool following l-p+r (where p is phoneme, l & r are left & right context) structure for each phoneme 'p' in monophones model making it context dependent e.g. in word Bharat, for phoneme 'a' triphones generated is bh-a+r.

Re-estimated monophones model obtained using HMM Toolkit:-

```
~h "a"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 39  -6.793139e+00 -2.883674e+000 -1.007505e+001
<VARIANCE> 39  2.777049e+001 7.315862e+001 4.440626e+001 .....
<GCONST> 1.312633e+002
<STATE> 3
<MEAN> 39  -6.793139e+000 -2.926306e+000  -1.007505e+001 -
4.524404e-001....
<VARIANCE> 39  2.777049e+001 7.315862e+001  4.440626e+001....
<GCONST> 1.312633e+002
<STATE> 4
<MEAN> 39  -6.793139e+000 --1.007505e+001 -4.524404e-001.....
<VARIANCE> 39  2.777049e+001  3.455022e+001 5.118520e+001.....
<GCONST> 1.312633e+002
<TRANSP> 5  0.000000e+000 1.000000e+000 0.000000e+000
0.000000e+000 3.000000e-001  0.000000e+000 0.000000e+000
<ENDHMM>
```

Figure 4. Monophone hmmdefs File

After making triphones, Decision tree state tying is performed by running HHEd tool of Hidden Markov Model Toolkit. HHEd is used to cluster the states and then each cluster is tied. Decision trees are based on asking questions about the left and right contexts of each triphones and find those contexts which make the largest difference to the acoustics and which should therefore distinguish clusters .[6]

Then we used edit script tree.hed, which contains the instructions regarding which contexts to examine for possible clustering and the questions (QS) defined by user according to language.

```
    QS  "R_NonBoundary"          { *+* }

    QS  "R_Stop"                 {*+p,*+ph,*+b,*+t,*+d,*+dd,*+k,*+g,
    *+dh,*+kg}

    QS  "R_Nasal"                  { *+m,*+n }

    QS  "R_Unvoiced-All"         {*+p,*+t,*+tt,*+k,*+kh,*+tth,*+chh,
*+ch,*+sil }

    QS  "R_NonAffricate"          { *+s,*+sh,*+z,*+f,*+v,*+th,*+dh }

    TB 100 "ST_sil_2_"           {("sil","*-sil+*","sil+*","*-sil").state [2]}

    TB 100 "ST_bh_3_"            {("bh","*-bh+*","bh+*","*-bh").state
```

Figure 5. Tree.hed File

Decision tree clustering of states is performed by TB commands.

Re-estimated triphones model obtained using Hidden Markov Model Toolkit :-

```
~h "n-d+a" <BEGINHMM>
 <NUMSTATES> 5
<STATE> 2
<MEAN> 39  -6.793139e+000 -2.926306e+000 -7.762902e+000…..
<VARIANCE> 39  2.777049e+001 4.440626e+001 5.118520e+001 ….
<GCONST> 1.312633e+002
<STATE> 3
 <MEAN> 39  -6.793139e+000 -1.007505e+001 -4.524404e-001…..
<VARIANCE> 39  2.777049e+001 7.315862e+001 4.440626e+001
5.118520e+001…..
<GCONST> 1.312633e+002
<STATE> 4
<MEAN> 39  -6.793139e+000  -7.762902e+000 -2.883674e+000 -
1.007505e+001 -4.524404e-001….
<VARIANCE> 39  2.777049e+001 7.315862e+001 4.440626e+001
6.700849e+001….
 <GCONST> 1.312633e+002 ~t "T_d"
<ENDHMM>
```

Figure 6. Triphone hmmdefs File

After decision tree clustering following file is obtained:

```
~h "a" <BEGINHMM>
 <NUMSTATES> 5
<STATE> 2 ~s "ST_a_2_1"
<STATE> 3 ~s "ST_a_3_1"
 <STATE> 4 ~s "ST_a_4_1" ~t "T_a"
<ENDHMM>
```

Figure 7. Clustered hmmdefs file

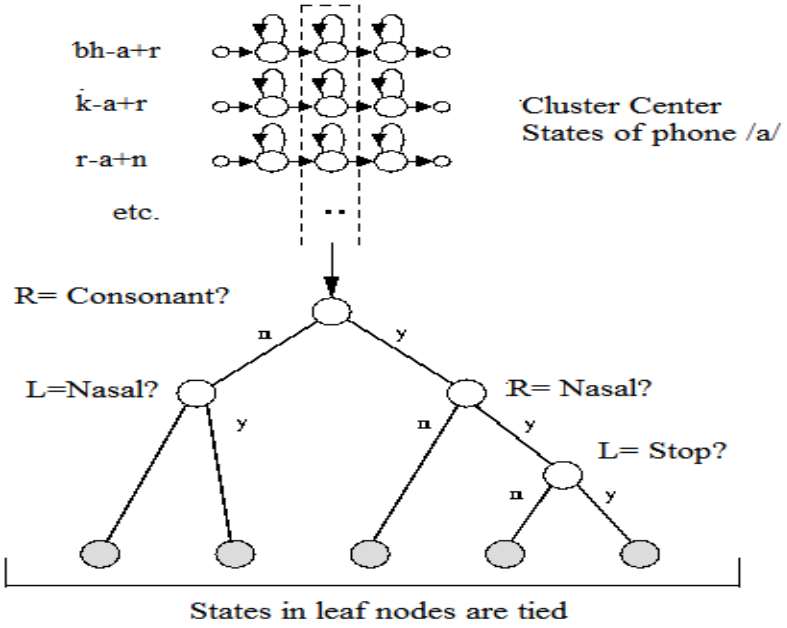


Figure 8. Decision Tree Clustering

3.4 Test Data Decoding

In this part the test data i.e. text which is to be converted to speech signals is given as input along with re-estimated context-dependent Hidden Markov Models obtained after training phase. According to the phoneme sequence in text labels the context-dependent Hidden Markov Models are concatenated with the help of HVite tool and word network file wdnet. According to the obtained state, the sequence of Mel cepstral coefficients and log F0 values including voiced/ unvoiced decisions are determined by maximizing the output probability of Hidden Markov Model [3]. Hidden Markov Model Toolkit is analyzed and used to find the appropriate pronunciation of a word from several alternate pronunciations of the words containing ਤ and ਟ i.e. whether Tony corresponds to ਤੋਨਿ or ਟੋਨਿ and words containing ਅ or ਆ and corresponding phonemes will be generated.

4. SPEECH SYNTHESIS RESULTS

The conflicting words will be dealt in phased manner. Initially to begin with, in the testing phase, Text-To-Speech data include:

Test-I : 28 Punjabi words with ਤ or ਟ and 45 Punjabi words with ਅ or ਆ

Test-II: 33 Punjabi words with ਤ or ਟ and 36 Punjabi words with ਅ or ਆ

The text labels are transformed into triphones format with the help of HMM Toolkit. For each word we have recorded a wav file i.e. bharat.wav etc. Appropriate pronunciation is selected from the network of alternate pronunciations.

A MLF (Master Label File) phonemes.mlf was generated by HLed tool that contained the correct phoneme sequence, amongst various other alternatives, of the test word contained in recout.mlf file which was initially produced by HVite tool of HMM Toolkit by making use of dictionary dict and word network file wdnet. Phonemes of different test words were arranged in different .lab files according to the input test samples.

Through HMM Toolkit correct sequences of phonemes is generated to a greater extent and satisfied results are obtained. Further rule based approach is used to formulate certain rules for generating phoneme sequences of words whose sequences are not correctly produced by HMM Toolkit.

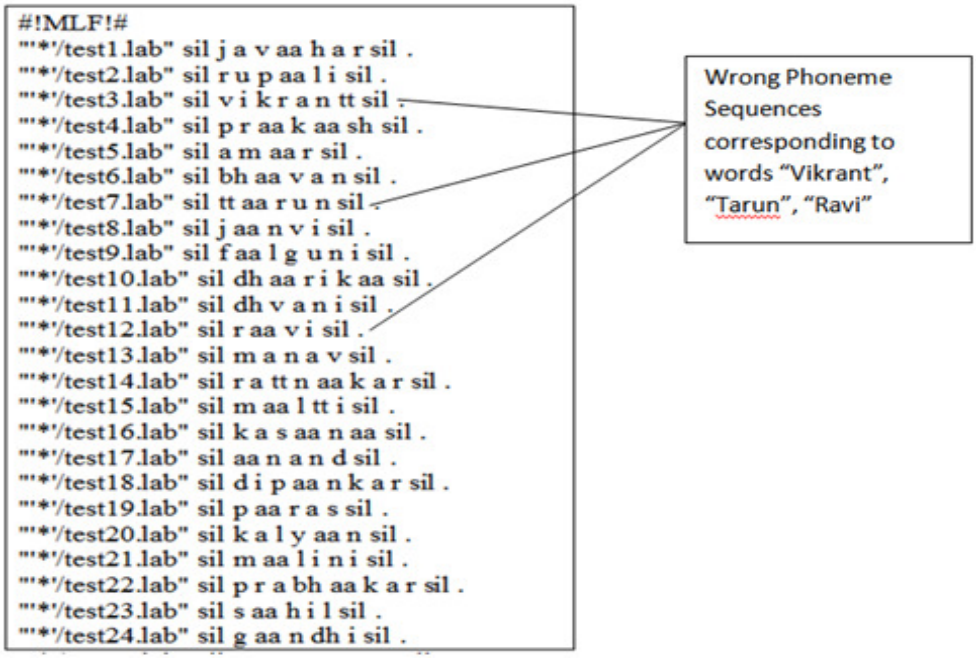


Figure 9. (phonemes.mlf) Phoneme File Generated

After testing HTK for various test samples having words containing ਤ or ਟ and ਅ or ਆ for which system is trained in training phase following results are obtained:-

- Both Test-I with 45 test words and Test-II with 36 words (containing ਅ or ਆ) and words obtained, after testing, with correct and incorrect phoneme sequences are represented by following bar graphs.

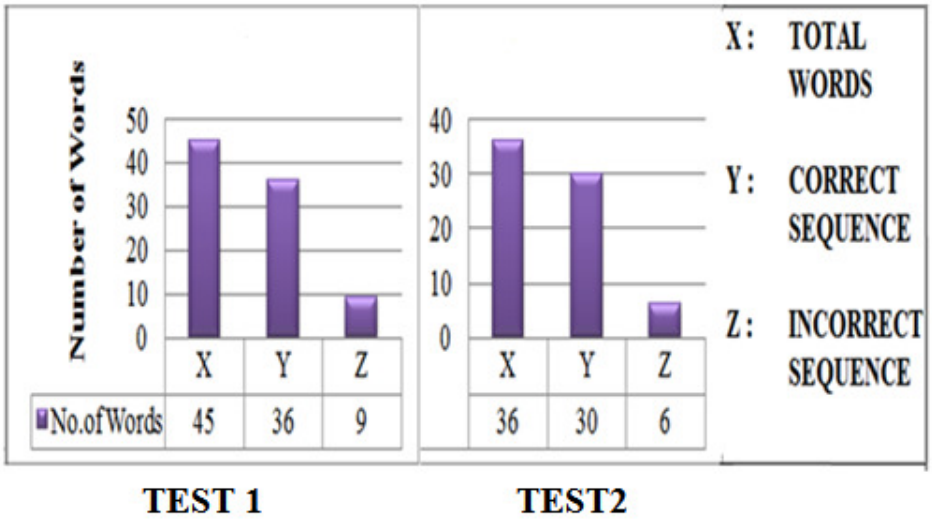


Figure 10. Test Samples for words containing ਅ or ਆ

- Number of words in Test Samples containing ਤ or ਟ with total 28 and 33 words and no of correct and incorrect phoneme sequences obtained are shown by following bar graph.
-

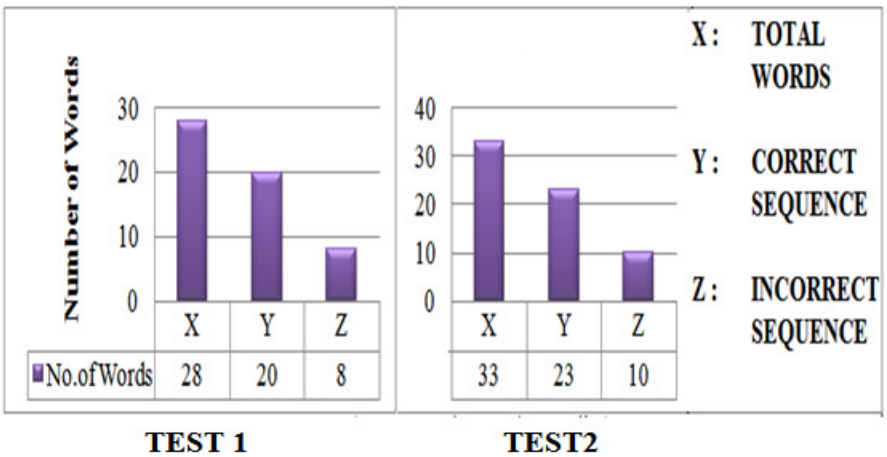


Figure 11. Test Samples for words containing ਤ or ਟ

- Comparisons of overall accuracies for both sets of test i.e. one containing words with ਤ or ਟ and one with ਅ or ਆ is depicted in following cylindrical bar graph.

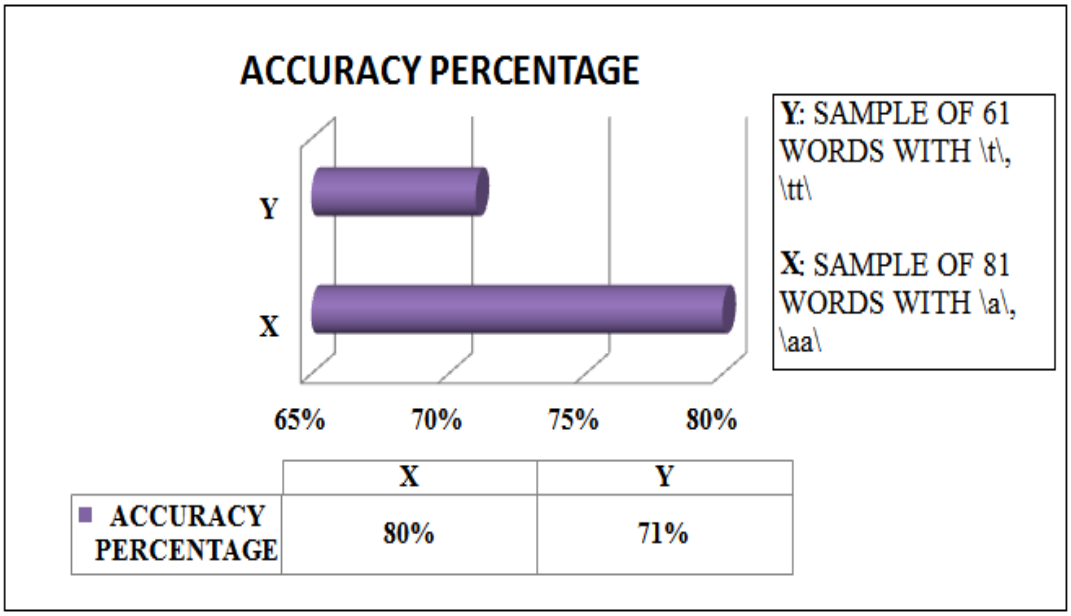


Figure 12. Comparison of Overall Accuracies

Figure 13 presents the result of generated speech for the sentences using Matlab code mfcc2spectrum [10].

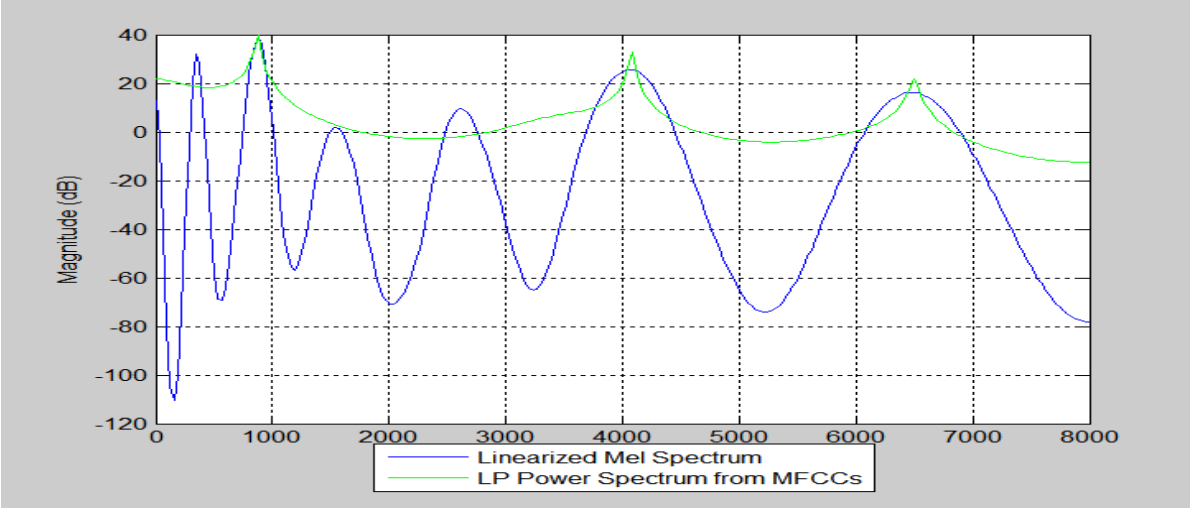


Figure 13. Spectrum representation from Mel Cepstral Coeff. data of word TONY

5. DISCUSSION AND CONCLUSION

HMM-based Punjabi speech synthesis system is presented in this paper. The developed Text-to-Speech was trained in phase -I on 17 samples with total 61 words all starting with letter ਤ and ਟ and tested for selection of appropriate phoneme sequence on 30 Punjabi words in test 1 and trained for 23 samples containing 81 words containing ਅ and ਆ and tested for 45 selected words in corresponding test-1. Hidden Markov Model Text-to-Speech system approach is very effective for developing Text-to-Speech systems for various languages and can easily implement changes in voice characteristics of synthesized speech with the help of speaker adaptation technique developed for speech recognition [7]. In order to improve efficiency, context-dependent phone models used for synthesis need to be improvised by recording, annotating more Punjabi speech data and applying filters using custom rules/ procedures.

REFERENCES

[1] S.D.Shirbahadurkar and D.S.Bormane, (2009) "Marathi Language Speech Synthesizer Using Concatenative Synthesis Strategy (Spoken in Maharashtra, India)", Second International Conference on Machine Vision, pp. 181-185.

[2] S. Martincic-Ipsic and I. Ipsic, (2006) "Croatian HMM-based speech synthesis," Journal of Computing and Information Technology, Vol. 14, no. 4, pp. 307–313.

[3] D. H. Klatt, "Review of Text to Speech Conversion for English", Journal of the Acoustic Society of America, 1987. Vol. 82, pp. 737–793.

[4] K. Tokuda, et al. (2003), "Multi-Space Probability Distribution HMM", IEICE Trans. Inf. & System, Vol. E85-D, No.3, pp. 455-464.

[5] L. R. Rabiner, (1989), "A tutorial on hidden Markov models and selected applications in speech recognition", Proc. IEEE, Vol. 77, No. 2, pp. 257–286.

[6] S. Young, et al. (2002), "The HTK Book (for HTK Version 3.2)", Cambridge University Engineering Department, Cambridge, Great Britain.

[7] K. Tokuda, H. Zen, A.W. Black, (2002), "An HMM-based speech synthesis system applied to English", Proc. of 2002 IEEE Workshop in Speech Synthesis.

[8] K. Kumar and R. K. Aggarwal, (2011), "Hindi Speech Recognition System Using HTK", International Journal of Computing and Business Research, Vol. 2, no. 2, pp. 2229-6166.

[9] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, (1999), "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in Proc. Eurospeech, pp. 2347–2350.

[10] N. Meseguer, "Speech Analysis for Automatic Speech Recognition", Master's thesis, Norwegian University of Science and Technology, Norway.

[11] S. King, (2011), "An introduction to statistical parametric speech synthesis", Sadhana - Engineering Science, Vol. 36 no. 5, pp. 837-852.

[12] P. Singh, (2005), Development of A Punjabi Text-To-Speech Synthesis System, M.Tech Thesis, Punjabi University.

[13] P. Singh and G. S. Lehal, (2006), Text-to-Speech Synthesis system for Punjabi language, Proceedings of International Conference on Multidisciplinary Information Sciences and Technologies, pp. 388-391.

[14] P. Gera, (2006), Text-To-Speech Synthesis for Punjabi Language, M.Tech Thesis, Thapar University.

Authors

Khushneet Jindal received his Post Graduation degree in M.Tech.(Information Technology) from KoSU, Master of Computer Applications from Punjabi University, Patiala, Punjab, India and Graduation in BSc.(Computer Applications) from Khalsa College Patiala, Punjab, India. His research interests are in the area of Speech analysis and processing, Character Recognition & TTS.

Divya Bansal received her graduation degree in Computer Science and Engineering from Punjab Technical University and is currently doing her post graduation in Computer Science and Application at Thapar College of Engineering and Technology, Patiala, Punjab, India. Her research interests are in the area of Speech analysis and processing.
\

Ankita Goel graduation received her graduation degree in Information Technology from Indraprastha University and is currently doing her post graduation in Computer Science and Application at Thapar College of Engineering and Technology, Patiala, Punjab, India. Her

.