Hidden Markov Model (HMM) based Speech Synthesis for Urdu Language

Omer Nawaz
Centre for Language Engineering, Al-Khawarizmi Institute of Compute Science, UET, Lahore, Pakistan.
omer.nawaz@kics.edu.pk

Dr. Tania Habib
Computer Science and Engineering Department UET, Lahore, Pakistan.
tania.habib@uet.edu.pk

Abstract

This paper describes the development of HMM based speech synthesizer for Urdu language using the HTS-toolkit. It describes the modifications needed to original HTS-Demo-scripts to port them, for Urdu language, which are currently available for English, Japanese and Portuguese. That includes the generation of the full-context style labels and the creation of the Question file for Urdu phone set. For that the development and structure of utilities are discussed. Plus a list of 200 high frequency Urdu words are selected using the greedy search algorithm. Finally the evaluation of these synthesized words is conducted using naturalness and intelligibility scores.

Keywords— Speech Synthesis, Hidden Markov Models (HMMs), Urdu Language, Perceptual Testing

1. Introduction

A text-to-speech (TTS) synthesis system for a particular language is a framework to convert any given text into its equivalent spoken waveform representation. Currently the most frequently employed TTS is the Unit Selection Synthesis [1-3]. However being the best TTS to date it has some limitations. Like the synthesized speech resembles the prosody/style of recording with the training database. If we want to synthesize speech with various voice characteristics then we need to increase the training data that cover all that variations. However recording that much data is not feasible [4]. With the improvements in Hidden Markov Models (HMM) techniques, the HMM based speech synthesizers are becoming popular [5]. In these systems the statistical models are trained based on source filter model from the training corpus. The main advantage of parametric approach [6] is that original waveforms are not required to be stored for synthesis purposes. As an effect the foot-print[1] is very small (approximately 2-MB[2]), compared to unit selection approach.

The HMM-based speech synthesis framework has been applied to a number of languages that include English [7], Chinese [8], Arabic [9], Punjabi [10], Croatian [11] and Urdu [12] as well. In this work, we present the development and evaluation of Speech Synthesizer for Urdu language. The main contributions of the paper are inclusion of prosodic information in the training process and development of question set considering the linguistic features relevant to Urdu language.

Figure 1 depicts the outline of parametric speech synthesis with HMMs. The training part consists of extracting the feature vectors of the training corpus as mel-cepstral coefficients [13] and excitation parameters, followed by model training. Whereas synthesis part is the reverse process of speech recognition. First the text is converted to context dependent sequence of phones obtained as a part of Natural Language processing (NLP) [14]. Then the excitation and spectral parameters are obtained through a set of trained HMM models using parameter generation algorithm [15]. Finally the waveform is generated using the obtained spectral and excitation features and providing them to the mel-log spectrum approximation filter (MLSA) [16].

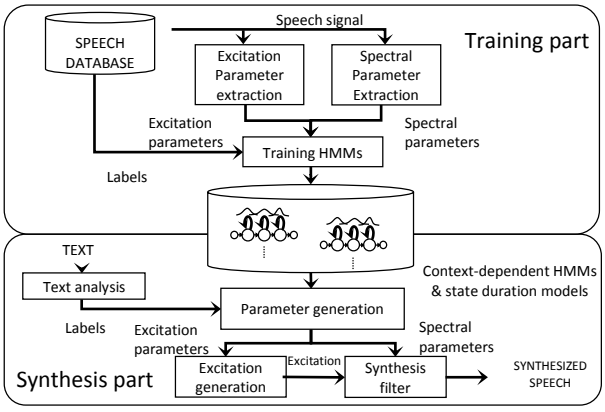

Figure 1. Overview of Parametric speech synthesis with HMMs ([7], pp. 227)

[1] Footprint refers to the amount of disk space required by an application.

[2] This refers to the voice size produced by the HTS-English-Demo Scripts.

1.1. Development

The development consists of two steps, training and synthesis. In the training step the recorded data, along with segmental and prosodic labels, is used to train the HMM models. The HMMs are trained on speech features that include MFCCs, F_0 and durations. Whereas in synthesis stage first the text to be synthesized is converted into a sequence of context dependent label format. This label structure contains the segmental and prosodic information that is helpful in selecting the appropriate models for the synthesis purpose. Finally the selected speech parameters are passed to the synthesis filter to produce the waveform.

In this paper we present the development of HMM based Speech Synthesizer for Urdu language and its evaluation. In section 2 we describe the requirements for training the HMM models, that include data collection, configuring influential features and the generation of the question file to handle data sparsity issues. Section 3 represents evaluation process and results on the test data. Section 4 encompasses analysis and discussion of the results. Finally section 5 discusses the concluding remarks and our plans for future.

2. Requirements for building Speech Synthesizer with HMM based Speech Synthesis Toolkit (HTS)

HTS is a toolkit [17] for building statistical based Speech Synthesizers. It is created by the HTS-working group as a patch to the HTK [18]. The purpose of this toolkit is to provide research and development environment for the progress of speech synthesis using statistical models.

The requirements for setting up the synthesizer are:

1. Annotated Training data.
2. Define speech features (MFCC, F_0 and duration) for model training.
3. Sorting out unique context-dependent as well as context-independent phonemes (from the training data) for model training.
4. Unified question file for spectral, F_0 and duration for context clustering.

2.1. Annotated Training data

For the system development, 30-minutes of speech data was selected. The recorded utterances consisted of paragraphs taken from Urdu Qaida of grade 2 and 4 respectively. The recordings were carried out in an anechoic room ensuring minimal noise and using a high quality microphone. The data was recorded by a native female speaker and stored at a sampling rate of 8 kHz mono wav format.

2.1.1. Importance of Segmental and Prosodic labels. Segmental (Phoneme) boundaries are required in continuous speech to identify the different phones present in the training data. The segmental labels are marked carefully by a highly trained team of Linguistics at CLE using Praat [19] software and saved in its native TextGrid format. The other marked layer is the word layer, which specifies the word boundaries. By having the word layer marked explicitly, we can apply 'stress' and 'syllabification' rules to it and can generate additional two layers.

With the addition of extra layers the advantage is that now we have more information, and can represent a single phoneme in a number of different contexts, which is important because the characteristics of a certain phoneme are greatly influenced by its context.

For the addition of extra layers (stress/syllable) a utility was written in Python [20], to mark the layers in TextGrid [19] format. The functionality of the program is explained:

Input: Take TextGrid file with segmental and word layer.

Process:

1. Extract different layers (currently possible Segment, Word, Stress, Syllable and Intonation).
2. Apply Stress/Syllabification rules [21].
3. Align stress and syllable identities with the segment layer.

Output: Generate a new TextGrid file

The block diagram of the utility is shown in Figure 2.

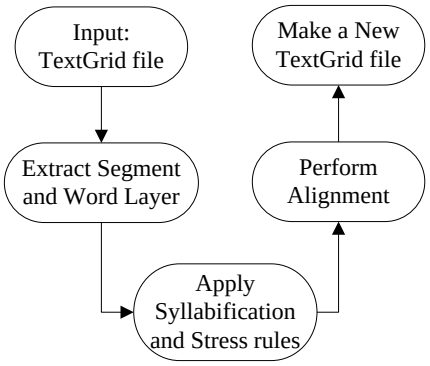

Figure 2. Block diagram of Stress/Syllable marking utility

Next the generated TextGrid file is converted to HTK-format for further processing, because HTS is implemented as a modified version of HTK. And HTK modules require labels in its native format.

2.1.2. Conversion to context-dependent label format. HTS requires both the basic HTK and its extended version 'Context-Dependent' to capture the prosodic variation of the phoneme.

In basic HTK-format each phoneme is represented by a string identity and time values are represented in units of 100 ns interval as shown in the example below:

HTK-Label Format

[Start	**[end]]**	**name**
0000000	3600000	**A_A**

With 'Context-Dependent' format the phoneme identity exists in different segmental plus supra-segmental contexts as shown below:

[Start	**[end]]**	**name**
0000000	3600000	SIL^S-**A_A**+L=SIL

@1_1/A:0_0_0/B:0-x-1@1-1&x-x#x-x$x-x!0-0;x-x|A_A/C:x+0+x/D:0_0/E:x+1@x+x&x+x#x+x/F:0_x/G:0_0/H:x=x^1=1|NONE/I:0_0/J:1+1-1

Supra-Segmental Context

Segmental Context

2.1.3. Phone Set Used. The CISAMPA phone-set [22] is employed in our system, which uses an ASCII based character set to represent different phonemes. It was chosen because these characters are easily accessible during the data tagging process.

For the conversion of TextGrid to 'Context-Dependent' format a utility was developed. The flow of this utility is illustrated in Figure 3:

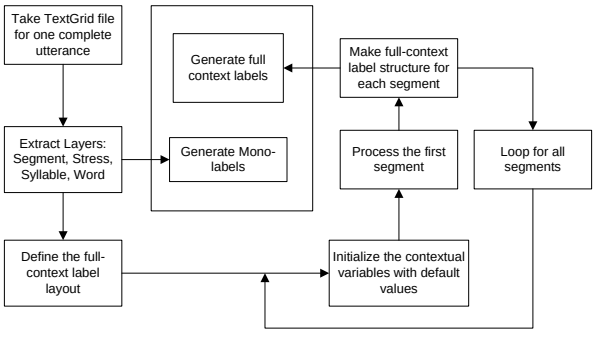

Figure 3. Flow diagram of the HTS-format conversion utility

i. TextGrid File

Take TextGrid file for one complete utterance. Usually one complete utterance consists of a single sentence.

ii. Extract Layers

Different layers (segment, stress, syllable, word) are extracted, that will be used to calculate contextual factors for each phoneme.

iii. Mono-labels

The HTK-style labels are generated at this stage.

iv. Full-context layout

A general layout is defined for the HTS-format that will be used to incorporate contextual factors [23]:

(The details of this layout can be seen in "*lab_format.pdf*" file bundled in the HTS-Demo Scripts)

p1^p2-p3+p4=p5@p6_p7/A:a1_a2_a3/B:b1-b2-b3@b4-b5&b6-b7#b8-b9$b10-b11!b12-b13;b14-b15|b16/C:c1+c2+c3/D:d1_d2/E:e1+e2@e3+e4&e5+e6#e7+e8/F:f1_f2/G:g1_g2/H:h1=h2^h3=h4|h5/I:i1_i2/J:j1+j2-j3

v. Initialize

Initialize factors related to stress, syllable, phoneme and word with default values ('0' or 'x'). The layout is kept general with all possibilities so that at a later stage when additional layers are added, then little modification is required.

vi. Process 1st Segment

Selects the first segment and calculates all the possible contexts available to it.

vii. Make structure

With the calculated contextual factors in the previous step a structural representation is created using the defined layout.

viii. Generate Full-Context

A full context label file is generated that consists of full-contextual representation of every phoneme in the entire utterance.

2.2. Feature Specification

In HTS a set of noticeable speech features are selected which are important to capture speech variations properly, and through their proper manipulation high quality speech can be synthesized. The most common set of features are spectrum, F_0 and duration.

2.2.1. Spectrum. For spectrum mel-cepstral coefficients (MFCCs) of order 35 were calculated along with the $\boldsymbol{\Delta}$ and $\boldsymbol{\Delta}^2$ features, so the final length is 105. For

the calculation of these features the SPTK toolkit [24] was used.

2.2.2. Fundamental frequency (F_0). The range defined for the voiced regions was 80-400 Hz. Fundamental frequency was calculated using the auto-correlation method used by the 'snack' library available in ActiveTcl [25].

2.2.3. Durations. There are two possible ways to estimate duration of each phone. One way is to calculate it offline like MFCCs and F_0, the other way is to estimate during the training process. The offline method is perfect if we only have a single state HMM. Whereas in most of cases we have 3 or 5 state HMM model, as a result we don't know in advance how the state alignments will be done. So, the durations of each state are estimated during the state-alignment step of the Expectation Maximization (EM) Algorithm which is used to train the HMMs.

2.3. Unique List

A list of unique context-dependent as well as context-independent phones from the training corpus is generated. It is required to identify number of possible models that can exist. Each model is created and trained with the available number of examples in the training corpus. First a context-independent (mono-phone) model is generated, which is simply an average of all the examples. Then these mono-phones are copied to context-dependent (full-context) and are re-estimated using the concerned examples only.

2.4. Question File

Some sort of criteria is required to tackle the problem of fewer training examples available per model. The numbers of training examples are few because if we look at the full-context style label format, then it reveals that the possible contextual occurrence of a single phoneme is quite huge. And to have this much context available in the training data is not possible, on the other side this much context is also rare in everyday speech.

To address these issues, a methodology known as clustering is employed. The notion of clustering is to group the phonemes which are acoustically similar and share a single model for closely related contexts. In clustering question file plays an important role, as they define how the grouping should be done.

The question file consists of a number of binary questions with YES or NO outcome related to segmental and prosodic context of the phoneme. It is helpful in the clustering process for spectrum, F_0 and duration models [26]. It provides a basis for grouping a number of data points and hence handles data sparsity issues, which is common in speech synthesis as the number of unique models built are enormous.

In clustering first all the data points are placed into a single cluster and a list of questions is made. Then an objective function is defined. The cluster is split based on each question, and the question with which the objective function is minimized is selected as a successful candidate and is removed from the list. And the remaining questions are asked on the resultant clusters until a stopping criterion is met [27].

Another advantage is that these trees can also be used in the synthesis stage, as they are built on the acoustically similar properties of speech. In synthesis mostly the utterance that is required to be synthesized is unseen. Meaning that utterance was not available exactly the same way in the training data. So, by using the trees that incorporate the acoustic similarities we can trace them and select the closest alternative.

2.4.3. Generation of the Question file. The questions are developed based on the similarities between the place and manner of articulation for segmental context. For prosodic context, the number of syllables in a word, their position and whether stressed or not etc. is taken into account.

The idea is to group the phones that have a similar place and manner of articulation. These set of questions have a dual role. In the training process they are used to split the cluster nodes of the tree. Whereas during synthesis process, these generated trees are employed to trace the phoneme with un-seen context.

Moreover these questions are created on a phone-set specific to language, we cannot employ the structuring specified for some other language (like English, Brazilian and Japanese). The questions specific to Urdu language were created considering the grouping in Table 1.

Question format

For example the question for phoneme before the previous phoneme is defined as:

Field 1	***Field 2***	***Field 3***
QS	"LL-TRILL_ALVEOLAR"	{R , R_H}

The field 1 defines the label showing that it is a question. Field 2 specifies the grouping of various categories and finally the third column represents the possible phonemes for the defined categories.

Table 1. Grouping of contextual factors[3]

VOWEL	All Vowels
CONSONANT	All Consonants
PLOSIVE	P, P_H, B, B_H, T_D, T_D_H, D_D, D_D_H, T, T_H, D, D_H, K, K_H, G, G_H, Q, Y
NASAL	M, M_H, N, N_H, N_G, N_G_H, U_U_N, O_O_N, O_N, A_A_N, I_I_N, A_E_N, A_Y_N
FRICATIVE	F, V, S, Z, S_H, Z_Z, X, G_G, H
VOICED, PLOSIVE	B, B_H, D_D, D_D_H, D, D_H, G, G_H
UNVOICED,PLOSIVE	P, P_H, T_D, T_D_H, T, T_H, K, K_H
VOICED, NASAL	M, M_H, N, N_H, N_G, N_G_H
UNVOICED, PLOSIVE	P, P_H, T_D, T_D_H, T, T_H, K, K_H
VOICED, FRICATIVE	V, Z, Z_Z, G_G
UNVOICED, FRICATIVE	F, S, S_H, X
PLOSIVE, ASPIRATED	P_H, B_H, T_D_H, D_D_H, T_H, D_H, K_H, G_H
AFFRICATES, ASPIRATED	T_S_H, D_Z_H
CONSONANT, ASPIRATED	P_H, B_H, T_D_H, D_D_H, T_H, D_H, K_H, G_H, M_H, N_H, N_G_H, R_H, R_R_H, L_H, J_H, T_S_H, D_Z_H
CONSONANT, ALVEOLAR	T, T_H, D, D_H, N, N_H, R, R_H, S, Z, L, L_H
CONSONANT, DENTAL	T_D, T_D_H, D_D, D_D_H
CONSONANT, VELAR	K, K_H, G, G_H, N_G, N_G_H, X, G_G
CONSONANT, BILABIAL	P, P_H, B, B_H, M, M_H
CONSONANT, UVULAR	Q
TRILL, ALVEOLAR	R, R_H
CONSONANT, LARYNGEAL	Y, H
FLAP, RETROFLEX	R_R, R_R_H
NASAL, ALVEOLAR	N, N_H
NASAL, BILABIAL	M, M_H
NASAL, VELAR	N_G, N_G_H
FRICATIVE, VELAR	X, G_G
FRICATIVE, PALATAL	S_H, Z_Z
FRICATIVE, LABIO-DENTAL	F, V
FRICATIVE, LARYNGEAL	H
LATERAL, ALVEOLAR	L, L_H
APPROXIMANT, PALATAL	J, J_H
FRICATIVE, ALVEOLAR	S, Z
AFFRICATES, PALATAL	T_S, T_S_H, D_Z, D_Z_H
VOWEL, FRONT	I_I, I, A_E, E, A_Y, I_I_N, A_E_N, A_Y_N
VOWEL, HIGH	U_U, I_I, U_U_N, I_I_N
VOWEL, HIGH-MID	U, I
VOWEL, LOW-MID	O, A_A, E, O_N, A_A_N
VOWEL, LOW	A_Y, A_Y_N
VOWEL, CENTRAL	A
VOWEL, BACK	U_U, U, O_O, O, A_A, U_U_N, O_O_N, O_N, A_A_N
SILENCE	SIL, BRTH, PAU

[3] The Urdu CISAMPA Phone set can be accessed at http://www.cle.org.pk/resources/CISAMPA.pdf

2.4.4. Synthesis. In the synthesis stage first the text to be synthesized is entered in CISAMPA format, then using the utilities developed in the training stage, converted to full-context style labels.

The label format used in the synthesis part is similar to the training, except for the timing information which is absent it this case. By using these labels three different set of models are selected (Spectrum, Fundamental frequency and the Durations). From Spectrum and Duration models the optimal state sequence is selected. Finally the optimal state sequence along with the excitation signal is fed to the synthesis filter to produce the final waveform, as illustrated in Figure 4.

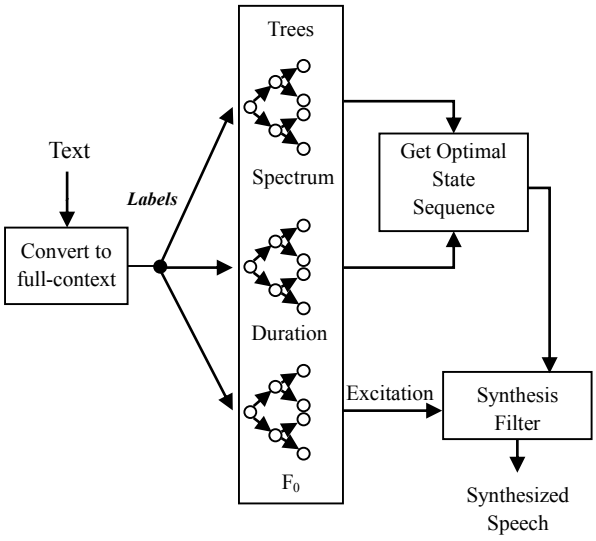

Figure 4. Overview of Synthesis Process

2.4.5. Tree traversal for model selection. For example if we want to synthesize the word *P A K I S T A N*, then for each phoneme a separate tree will be used to trace the appropriate model. In this example we may have two different models for the 'A' phoneme, because it is occurring twice and have different left and right contexts. For the case of '**P A K**', first it checks whether left context is bilabial or not (L=Bilabial?). Then on its outcome it proceeds to the next node, and finally reaches the leaf node from where suitable model is selected as shown in Figure 5.

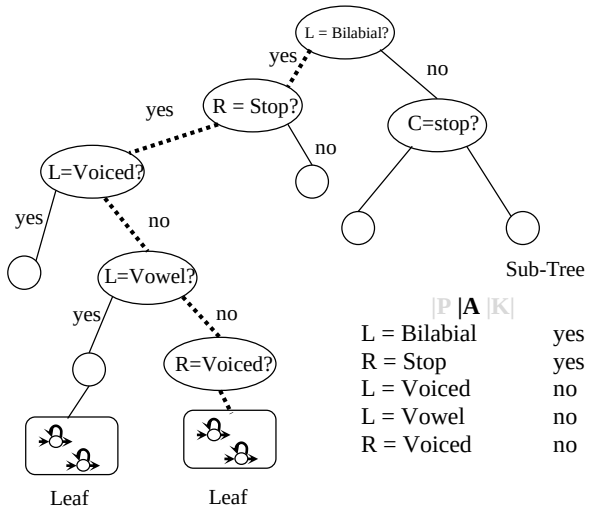

Figure 5. Tree traversal for model selection

3. Evaluation and results

The main goal of any Text to Speech System is to generate a voice which resembles closely to a human voice. So for the assessment of a speech synthesizer, a human listener should carry out the testing.

To test the system comprehensively, there are a number of tests, which include Diagnostic Rhyme Test (DRT) [28] and Modified Diagnostic Rhyme Test (M-DRT) [29] that evaluates the system on the phoneme level.

For our system we only focused on the naturalness and intelligibility of high frequency words. As Consonant-Vowel-Consonant (CVC) [30] or DRT was not appropriate because the number of possible correct words fitting exactly the CVC format was scarce. Moreover, we did not have the phoneme coverage balanced for the 30-minutes of the speech data. The phoneme coverage graph is shown in Figure 6. Consequently a set of 200 high frequency words were selected for the testing purpose.

3.1. Methodology

To perform evaluation of the underlying system a list of 200 high frequency words of Urdu language were selected using the greedy search algorithm [31] developed at Center for Language Engineering, KICS.

3.2. Experiment

For the assessment of speech quality synthesized by the statistical models (HMMs). The Mean-Opinion-Score (MOS) was considered for the naturalness and intelligibility measure. There were a total of 4 listeners who carried out the evaluation. Among the participants, three were linguists (expert listeners) and one was technical (naive listener).

For our system the naturalness and intelligibility are interpreted as:

Naturalness: How close it seems to be produced by a human?

Intelligibility: How much conveniently the word was recognized?

The MOS-scale varies from 1 to 5, where 1 represents the lowest score and 5 the highest. The experimental results of four listeners are listed in Table 2:

Table 2. Mean score for Intelligibility and Naturalness

Subject Type	MOS Naturalness	MOS Intelligibility
Technical 1	3.23	3.65
Linguistic 1	2.82	3.66
Linguistic 2	2.86	3.58
Linguistic 3	3.48	3.52

The testing reveals that most of the words were intelligible but not natural. The reason behind un-natural voice can be regarded due to the kind of training data. In training, words were available as a carrier sentence, and none of the training utterance consisted of a single word. We know that if a word is spoken explicitly without any carrier sentence then it is little bit longer and clearer, whereas in carrier some of the phonemes are shorter or are completely ignored.

4. Analysis and Discussion

The analysis show that on average **92.5%** words were correctly identified, irrespective they sounded less natural or intelligible. On the other hand, there were also a few cases where the listener was unable to identify the correct word. These are listed in Table 3.

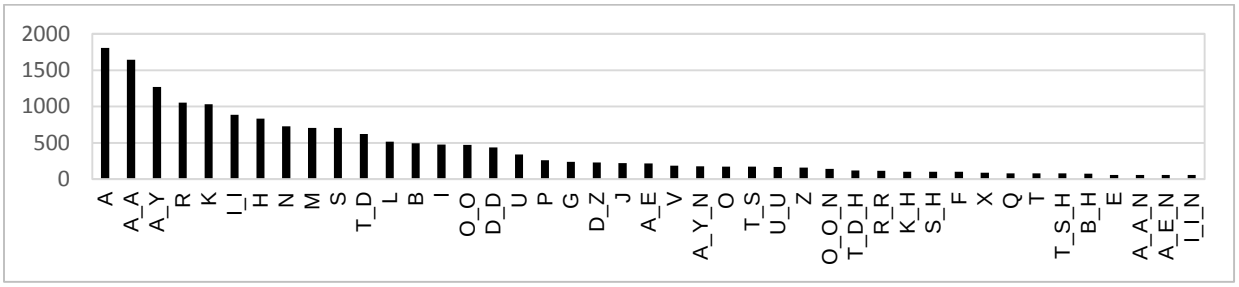

Figure 6. Phoneme Coverage Graph[4]

4.1. Phoneme coverage of training data

There were total of 66 different phonemes present in the phone set defined for training HMMs Models, having total frequency of 17793 (30-minute data). So for completely phonetically balanced system we should have at least 270 (**1.51 %** coverage) examples per phoneme. Whereas in our case vowels had a very high frequency (A = 1810, A_A = 1646), and some of the consonants were completely ignored (J_H, L_H, M_H, N_G_H, R_H, Y, Z_Z). The phoneme coverage can be visualized in Figure 6.

A list of words that were not correctly identified are listed in Table 3.The first column contains the word in Nastalique style. Second represents the actual pronunciation that should have been synthesized, in CISAMPA format. Whereas third column represents the word interpreted by the listener. The bold letters highlights the phones which have disagreements, while gray letters indicates that they were missing in the synthesized utterance. Finally the last column represents the coverage of the correct phoneme that was wrongly produced, in the training corpus.

Table 3. Words with errors

Nastalique Style	CISAMPA (Correct)	Listened (Incorrect)	Coverage (%)
طرف	T_DA***R***AF	T_DA***L***AF	5.92
گا	***G***A_A	***D_D***A_A	1.35
معلوم	MA**Y**LU_UM	MA**T_D**LU_UM	0.00
تھے	***T_D_H***A_Y	***T_S***A_Y	0.66
رزی	RA***Z***I_I	RA***D_D***I_I	0.88
ہوتی	***H***O_OT_DI_I	*H*O_OT_DI_I	4.68
کیونکہ	KI***U_U_N***KA_Y	***T_SU_N***KA_Y	0.15
حق	HA***Q***	HA***BS***	0.46
بعد	BA***Y***D_D	BA*Y*D_D	0.00
خیال	***X***AJA_AL	***FI***JA_AL	0.50

5. Conclusion and Future Work

A reasonably good quality[5] HMM based speech synthesizer for Urdu language has been developed. The utilities developed were unique as they converted hand-labeled TextGrid files directly to HTS-label format, without using any of the automatic data tagging software (like Sphinx **[**32**]**). The Question file was generated for the Urdu phone set, keeping in account the articulatory features of language. Finally the testing of the synthesized quality was carried out by using the Mean-Opinion-Score (MOS) for naturalness and intelligibility.

In future work we are planning to build the system incrementally with new database which comprises of approximately 10-hours of speech and is being recorded by a professional speaker

Acknowledgments

This work has been conducted through the project, Enabling Information Access for Mobile based Urdu Dialogue Systems and Screen Readers supported through a research grant from ICTRnD Fund, Pakistan.

References

[1] A. W. Black and P. Taylor, "CHATR: a generic speech synthesis system," in proc. of the 15th conference on Computational linguistics, Stroudsburg, PA, USA, 1994.

[4] Only those phonemes are shown whose occurrence counts are more than 50

[5] Some synthesized utterances can be accessed at: http://www.cle.org.pk/tts/sample

[2] R. E. Donovan and P. C. Woodland, "Automatic speech synthesiser parameter estimation using HMMs," in proc. of ICASSP-95, Detroit, Michigan, May, 1995.

[3] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in proc. of ICASSP-96, IEEE International Conference, Atlanta, Georgia, May, 1996.

[4] A. W. Black, "Unit selection and emotional speech," in proc. of INTERSPEECH, Geneva, September, 2003.

[5] Z. Heiga, T. Tomoki, M. Nakamura and K. Tokuda, "Details of the Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," IEICE transactions on information and systems, vol. 90, no. 1, pp. 325--333, 2007.

[6] H. Zen, K. Tokuda and A. W. Black, "Statistical parametric speech synthesis," Speech Communication, vol. 51, no. 11, pp. 1039-1064, 2009.

[7] K. Tokuda, H. Zen and A. W. Black, "An HMM-based speech synthesis system applied to English," in Speech Synthesis, 2002. Proceedings of 2002 IEEE Workshop, Santa Monica, California, IEEE, September, 2002, pp. 227-230.

[8] Y. Qian, F. Soong, Y. Chen and M. Chu, "An HMM-based Mandarin Chinese text-to-speech system," in Chinese Spoken Language Processing, vol. 4274, Springer Berlin Heidelberg, Singapore, December, 2006, pp. 223-232.

[9] O. Abdel-Hamid, S. M. Abdou and M. Rashwan, "Improving Arabic HMM based speech synthesis quality," in proc. of INTERSPEECH, Pittsburgh, Pennsylvania, USA, September 17-21, 2006.

[10] D. Bansal, A. Goel and K. Jindal, "Punjabe speech synthesis using HTK," International Journal of Information Sciences & Techniques, vol. 2, no. 4, July, 2012, pp. 57-69.

[11] I. Ipsic and S. Martincic-Ipsic, "Croatian HMM-based speech synthesis," CIT. Journal of computing and information technology, vol. 14, no. 4, December, 2006, pp. 307-313.

[12] Z. Ahmed and J. P. Cabral, "HMM BASED SPEECH SYNTHESISER FOR THE URDU LANGUAGE," in 4th International Workshop On Spoken Language Technologies For Under-resourced Languages, St. Petersburg, Russia, 2014.

[13] T. Fukada, K. Tokuda, T. Kobayashi and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in proc. of ICASSP-92., IEEE International Conference, San Francisco, California, March, 1992.

[14] H. Kabir, S. R. Shahid, A. M. Saleem and S. Hussain, "Natural Language Processing for Urdu TTS System," in Multi Topic Conference, 2002. Abstracts. INMIC 2002. International, IEEE, 2002, pp. 58-58.

[15] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in proc. of ICASSP'00, IEEE International Conference, Istanbul, Turkey, June, 2000.

[16] S. Imai, "Cepstral analysis synthesis on the mel frequency scale," in proc. of ICASSP'83, IEEE International Conference, Boston, Massachusetts, USA, April, 1983.

[17] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in proc. of Sixth ISCA Workshop on Speech Synthesis, Bonn, Germany, August, 2007.

[18] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey and others, "The Hidden Markov Model Toolkit (HTK) version 3.4," Cambridge University Engineering Department, December, 2006.

[19] P. Boersma and D. Weenink, "Downloading Praat for Windows," 10 September 2013. [Online]. Available: http://www.fon.hum.uva.nl/praat/download_win.html.

[20] G. van Rossum and others, "Python language website," World Wide Web: http://www.python.org, 2007.

[21] S. Hussain, "Phonological Processing for Urdu Text to Speech System," in Contemporary Issues in Nepalese Linguistics (eds. Yadava, Bhattarai, Lohani, Prasain and Parajuli), Linguistics Society of Nepal, Nepal, 2005.

[22] A. Raza, S. Hussain, H. Sarfraz, I. Ullah and Z. Sarfraz, "An ASR System for Spontaneous Urdu Speech," in proc. of Oriental COCOSDA, Kathmandu, Nepal, November, 2010.

[23] H. Zen, "An example of context-dependent label format for HMM-based speech synthesis in English," The HTS CMUARCTIC demo, July, 2011.

[24] S. Imai, T. Kobayashi, K. Tokuda, T. Masuko, K. Koishida, S. Sako and H. Zen, "Speech signal processing toolkit (SPTK)," 2009.

[25] ActiveState and ActiveTcl-User-Guide, "Incr Tk", ActiveTcl 8.4.1.0, Nov, 2002.

[26] S. J. Young, J. J. Odell and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in proc. of the workshop on Human Language Technology, Plainsboro, New Jerey, USA, March, 1994.

[27] K. Shinoda and T. Watanabe, "Acoustic Modeling Based on the MDL Principle for speech recognition," in proc. of EuroSpeech-97, Rhodes, Greece, September, 1997.

[28] W. D. Voiers, "Diagnostic evaluation of speech intelligibility," Benchmark papers in acoustics, vol. 11, Stroudsburg, Pennsylvania, 1977, pp. 374-387.

[29] A. S. House, C. E. Williams, M. H. Hecker and K. D. Kryter, "Articulation-Testing Methods: Consonantal Differentiation with a Closed-Response Set," The Journal of the Acoustical Society of America, vol. 37, no. 1, January 1965, pp. 158-166.

[30] U. Jekosch, "The cluster-based rhyme test: A segmental synthesis test for open vocabulary," in Speech Input/Output Assessment and Speech Databases, Noordwijkerhout, 1989.

[31] B. Bozkurt, O. Ozturk and T. Dutoit, "Text design for TTS speech corpus building using a modified greedy selection," in INTERSPEECH, 2003.

[32] "CMU Sphinx - Speech Recognition Toolkit," Carnegie Mellon University, [Online]. Available: http://cmusphinx.sourceforge.net/html/cmusphinx.php. [Accessed 3 March 2014].