HMM-BASED SPEECH SYNTHESISER FOR THE URDU LANGUAGE

Zeeshan Ahmed[1], João P. Cabral[2]

CNGL, University College Dublin[1], Trinity College Dublin[2]

ABSTRACT

This work presents H idden Markov Model (HMM) based speech synthesis for the Urdu language. This is a widely spoken language across different regions in Asia. For example, Urdu is the official language of Pakistan and one of the national languages of India. Unfortunately, there is no corpus of Urdu currently publicly available that to our knowledge is appropriate for HMM-based speech synthesis purpose. We overcame this problem by recording an Urdu speech database with word and phone labels obtained using manual and semi-automatic annotation approaches. In summary, the objective of this work is to develop an HMM-based Urdu speech synthesiser from scratch by trying to use publicly available text processing tools for this language and by developing the necessary processing components.

Index Terms— HMM-based speech synthesis, Urdu speech synthesiser, Urdu speech corpus

1. INTRODUCTION

HMM-based speech synthesis [1] has gained great popularity in the last decades because it permits to build new voices using fully automatic techniques, it can produce high-quality speech and it offers high parametric flexibility for voice transformation. It is also an attractive approach for rapidly building systems for new languages with limited data and resources. The goal of this work is to build a statistical parametric speech synthesiser for a new language which is Urdu, understand the difficulties in building this new system and find out solutions to the problems encountered.

The main challenges for building HMM-based speech synthesisers for new languages are the availability of a sufficiently large speech database and text analysis tools for that language. The larger the database the better the speech quality that can be obtained and it usually ranges from 1 to 10 or more hours of recorded speech. A text analysis component is required for obtaining phonetic labels from text and to extract linguistic information, which is essential to model the linguistic contextual factors and to produce acceptable speech quality.

Urdu uses Arabic and Persian alphabets for scripting and shares several features of the pronunciation system with these languages. Similarly to these languages, the short vowels in Urdu are indicated in text using diacritics as shown in the example in Table 1. Processing diacritics is an important difficulty in the development of the Urdu speech synthesis system because these diacritics are often omitted. The effect of a slight error in prediction of a correct short vowel can result in huge variations of word semantics. For example, the word پر can be pronounced as پَر (per) which means “feather". It can also be pronounced as پُر (pur) which means “To fill"). The only difference between these two pronunciations is a short vowel, which is not indicated in the written form. For this reason, it is difficult for a synthesiser to generate the correct pronunciation from text if the diacritics are omitted. An accurate linguistic processor is needed to effectively avoid this problem in Urdu speech synthesis [2, 3].

Letter	Vowel name	English transcription	IPA
بَ	Zabar	ba	/bə/
بِ	Zer	bi	/bɪ /
بُ	pesh	bu	/bʊ /

Table 1: Diacritics on Urdu consonant ب.

The other great challenge for building the Urdu speech synthesiser is the availability of a suitable speech database for learning speech models. The voice building process in HMM-based speech synthesis requires speech recordings and their corresponding transcription at phone level. Currently, there is not a phonetically balanced speech corpus available for the Urdu language that can be used for HMM-based speech synthesis, to the best of our knowledge. Recently, efforts have been done towards the development of a phonetically rich Urdu speech corpus [4]. However, this data is not publically available. For this reason, it was necessary to collect a new speech corpus for Urdu in this work.

2. SYSTEM OVERVIEW

The HTS system [5] is a popular HMM-based speech synthesiser. Figure 1 shows the general structure of this system. It can be divided into the training and synthesis parts.

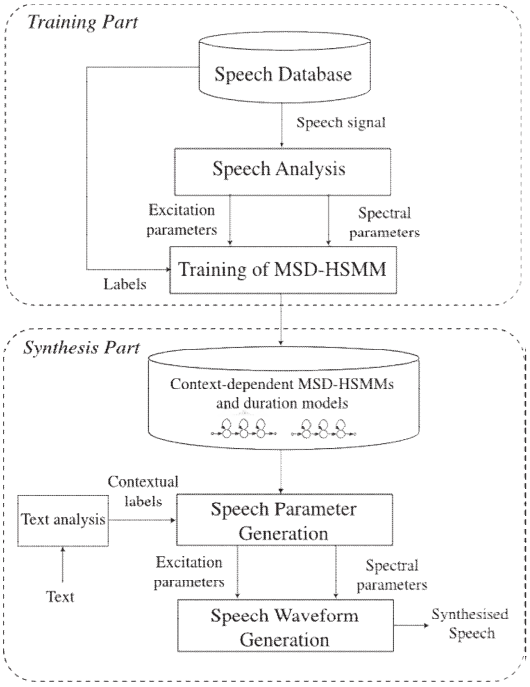

Figure1: General overview of the HTS architecture.

2.1 Training

At the training stage, phonetic, linguistic, and prosodic parameters are estimated from the sentences of the recorded speech corpus using text analysis tools. This information is represented by the HTS system in the form of "rich" linguistic labels which are used for training the context-dependent phone models. For example, the context information includes counts, positions and distances of stressed and accented syllables, as well as other context information at phone and utterance levels. The phone model is an extension of HMM for explicit duration modelling called Hidden Semi-Markov Model (HSMM). Speech parameters are also extracted for each utterance of the speech corpus. For example, the fundamental frequency F_0 is generally used as an excitation parameter, while mel-cepstral coefficients or line spectral frequencies are popular types of spectral parameters used for statistical modelling with HSMMs.

The next training step is to use the phonetic labels and the speech features to model context-dependent HSMMs. In this process the statistical parameters of the models are calculated using dynamic programming algorithms such as the Viterbi and forward-backward algorithms, e.g. [6]. The spectral parameters are modelled by a continuous probability distribution, e.g. Gaussian, because the spectral envelope is assumed to vary slowly across contiguous speech frames. In contrast, F0 is modelled using multi-space probability distribution (MSD-HSMM). MSD-HSMM permits to model F0 using a discrete distribution in unvoiced regions of speech (F0 values are not defined in these regions) and a continuous distribution in voiced regions (the F0 contour is approximately continuous in these regions).

The trained HSMMs are also clustered using decision trees which describe the contextual factors of the rich linguistic labels. In HTS, the spectral parameters, F_0 and HSMM state duration are clustered independently because they have their own influential contextual factors.

2.2 Synthesis

At the synthesis stage, the context-dependent labels are first obtained from the input text. Then, both the trained HSMMs and the input labels are used by the speech parameter generation algorithm to generate the speech features. The excitation signal is calculated using the excitation features, which then passes through the synthesis filter to obtain the speech signal. Meanwhile, this synthesis filter is defined by the spectral features.

3. SPEECH DATABASE

The phonetically rich corpus is a fundamental resource in the development of the speech synthesiser. In this work, the first author who is a native speaker of Urdu recorded his own speech and manually labeled the sentence boundaries. Speech was recorded using a microphone in a quiet room, at 16 kHz sampling frequency. The reading material was made up of articles from newspapers and magazines selected randomly. The database consisted of around one hour and 15 minutes of recorded speech, which corresponds to 989 sentences in total.

In addition to the manual annotation of sentence boundaries, a forced-alignment approach was used for obtaining phone level transcriptions using the HTK toolkit [7]. Figure 3 and 4 show examples of the annotation at the sentence and phone levels respectively.

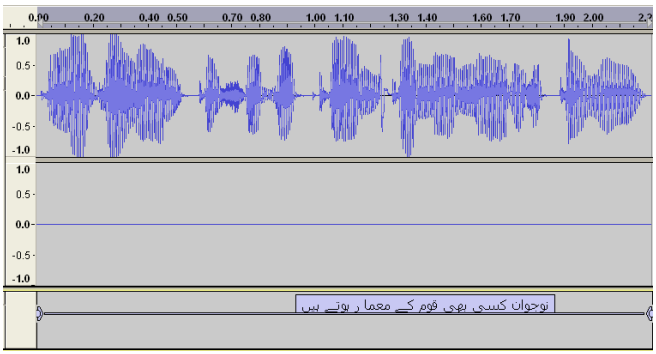

Figure 3: Aligned speech on sentence level.

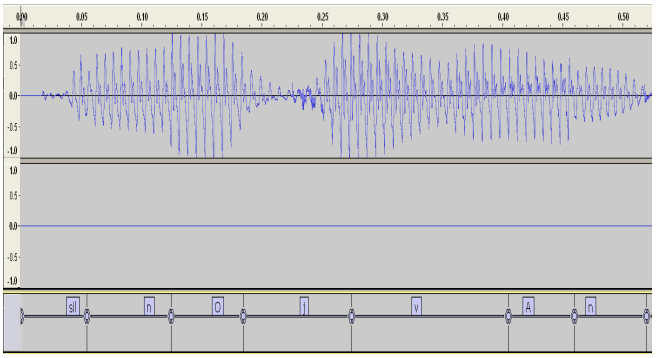

Figure 4: Force-alignment on phone level.

4. TEXT ANALYSIS COMPONENT

Different linguistic processing components are required for Urdu speech synthesis, in order to deal with different aspects of Urdu language [3]. The linguistic processor system developed for the HMM-based Urdu synthesiser in this work is composed of three major components which are represented in Figure 2.

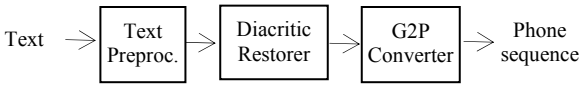

Figure 2: Components of the linguistic processor system for the Urdu HMM-based speech synthesiser.

The first stage of linguistic processing is the text preprocessing or normalization, such as tokenization, conversion of numerical symbols and abbreviations to full textual form. Next, the diacritic restorer is required for restoring diacritics (short vowels) in the Urdu language. Finally, a grapheme-to-phoneme (G2P) converter was developed for generating a sequence of phonetic sounds for a given sentence.

In text-to-speech synthesis it is also common to use other linguistic processing components. For example, the architecture of the linguistic processor for Urdu speech synthesis presented in [2] includes a syllabifier (for automatic segmentation of words into syllables). The integration of this component into our system is part of the plan for future work.

4.1 Text Preprocessing

An Urdu text tokeniser must deal with issues of word segmentation [8], identification of numerals, identification of punctuations, identification of date and time representations, etc. In this work, an n-gram based word segmentation technique was implemented as described in [8] for Urdu text segmentation. For numerals and date/time numerical notations, they are detected by the system in the text and processed using regular expressions while a dictionary is maintained for the punctuations.

A rule-based component was developed in this work for the conversion of numerals into textual form, because this type of conversion depends on the language and we could not find any available converter. For example, in English, numerals from 1-20 have distinct written forms and pronunciations, as well as for 100 (hundred), 1.000 (thousand), 1.000.000 (million) and 1.000.000.000 (billion), etc. In Urdu, there are also similar "basic numerals" which are written and pronounced differently. However, one of the differences between the two languages is that Urdu numerals are written and pronounced differently for basic numerals from 1-100 instead of the shorter range of 1-20 in English and for 1.000 (ہزار), 100.000 (لاکھ), 10.000.000 (کروڑ), etc. The written forms and pronunciations of numerals can be obtained using the combination of basic numerals. In our system, a dictionary is maintained for the written form of the basic numerals and a set of rules is used to derive the other numerals. For example, "156" is converted into "ایک سو چھپن", based on the rule that "1" is the first digit which is before two digits so it is converted into "ایک سو" while "56" is available in the dictionary because it is one of the basic numerals.

The abbreviation conversion is performed by direct mapping between the abbreviations and the respective words using a dictionary of abbreviations. Currently, the dictionary contains only the abbreviations found in the corpus created in this work.

4.2 Diacritic Restoration

A dictionary developed by the Centre for Research in Urdu Language Processing (CRULP) was used to implement the diacritic restoration component. This dictionary contains 80.000 Urdu words mapped to their diacritic version. The words of the recorded Urdu speech corpus which were not in this dictionary were manually added to the dictionary. Table 2 shows some examples of words from the dictionary and the respective diacritic variants.

Word	Word with Diacritics
آزمائش	آزْمائش
موجود	مَوجُود
تجزیہ	تَجْزِیَہ
جنت	جَنَّت

Table 2: Examples of words in the diacritic dictionary.

However, this dictionary-based approach cannot deal with the problem in which the diacritic of a word can be restored in multiple ways as explained in Section 1 for the example of the word "پر". In some cases, a part-of-speech (POS) tag can be helpful to solve this problem but semantic analysis may be required for solving more difficult problems characterized by higher ambiguity. In [9] a comprehensive discussion on the diacritics restoration in Urdu is presented and several possible solutions to solve this problem are proposed. According to this paper, a common solution is to use more context information. A similar approach was applied here in the form of bi-gram context.

4.3 G2P Converter

After the restoration of diacritics in the input text, the G2P conversion task is straight forward because Urdu has regular mapping between graphemes and phonemes, unlike English. However, there are some exceptions as explained in [3]. For example, the alphabet units ا (Alif), و (Vao), ے or ی (Yay) may occur either as a vowel or a consonant depending on the context. Context-based rules can be designed for mapping the alphabet units to vowel and consonants, as explained in [3]. The guidelines presented in [3] were used for developing the G2P conversion rules in this work.

The G2P converter was modeled using 36 basic consonants and 10 vowels. Details about the Urdu phonemic inventory can be found in [10], including discussion about the exact number of Urdu consonants and vowels. Some of the consonants in Urdu have their aspirated variants. For example, the /**p**$^{\mathbf{h}}$/ sound is the aspirated variant of the plosive /**p**/. In our system, aspirated variations are modeled using a separate /**h**/ sound i.e. /**p**$^{\mathbf{h}}$/ is modeled using a plosive /**p**/ and an aspiration /**h**/.

All the words in our speech corpus were transcribed into phonetic representation using the G2P converter and stored in a pronunciation dictionary. In addition, each word in the dictionary was checked by the first author and manually corrected for any discrepancies. The dictionary currently contains 3.673 words but we intend to further extend it in future work. During speech synthesis, the phone sequence of the pronunciation dictionary is given preference over the one generated by the G2P converter because the first is considered to be more accurate as it was manually checked.

5. URDU SPEECH SYNTHESISER

The Urdu synthesiser is based on the HTS-2.2 toolkit [5] and integrates the STRAIGHT vocoder [11], which is used to estimate the speech parameters during analysis and to generate the speech waveform at the synthesis stage.

5.1. Training

During speech analysis, STRAIGHT is used to extract the spectrum and aperiodicity parameters from the speech signal. Meanwhile, F0 is estimated using the RAPT algorithm [12] implemented in the Entropic Speech Tools [13]. The spectral parameters were obtained by converting the spectral envelope computed by STRAIGHT (defined by 512 FFT coefficients) to 39th mel-cepstral coefficients, which are better for statistical modelling by the HSMMs. The FFT coefficients defining the aperiodicity measurements were also converted into a lower number of parameters by averaging the measurements in five frequency bands: 0-1, 1-2, 2-4, 4-6, and 6-8 kHz.

For acoustic modelling, the system uses a five-state left-to-right HSMM structure. The F0 parameter vector (including its delta and delta-delta features) is modelled by multi-space probability distribution HSMM (MSD-HSMM), whereas the spectrum and aperiodicity streams (including dynamic features) are modelled by HSMM using continuous distributions respectively.

The F0, spectrum and aperiodicity parameters were clustered using different decision trees, because these parameters have their own contextual factors. The phone context questions used for tree-based clustering were defined by phonetic features of the International Phonetic Alphabet (IPA). These features were also represented in the Urdu phone set used to transcribe the corpus and are indicated in Table 3. The combinations of these features resulted in 680 questions for building the decision trees.

Class of sound	vowel / consonant
Vowel length	short / long /diphthong / schwa
Vowel height	high / mid /low
Vowel front	front / mid /back
Consonant type	stop / fricative / affricative / nasal / approximant / tap
Lip rounding	yes / no
Place of articulation	labial / alveolar / palatal / labio-dental / dental / velar / uvular / retroflex / glottal
Consonant voicing	voiced / unvoiced
Aspiration	yes / no

Table 3: Phonetic features of IPA used for context clustering of the HSMMs.

5.2 Synthesis

Labels with phonetic and linguistic context information are firstly generated from the input text using the Urdu Linguistic Processor. Then, the speech parameters are generated from the context labels and trained HSMMs using a parameter generation algorithm based on the maximum likelihood criterion. Finally, the speech waveform is produced from the speech parameters using the STRAIGHT vocoder.

Some examples of utterances generated by the Urdu HMM-based speech synthesiser are available at *https://www.scss.tcd.ie/~cabralj/web/hts-urdu.html*.

Based on informal listening of several utterances by the authors the synthetic voice sounds close to that of the original speaker, but synthetic speech rate sounds like it is higher (sounds faster) than that of the recorded speech, in general. This effect may be related to a problem in modelling speech duration in the system, either due to the small size of the speech database or due to the limited contextual information used for acoustic modelling HSMMs, such as the limitation of not using syllable information as context features. We are currently investigating this problem. For example, we are going to test if by the inclusion of syllable information this effect will be reduced.

6. CONCLUSION

An HMM-based speech synthesiser for Urdu has been developed in this work. We were interested in rapidly building of the synthetic voice for the new language, with limited speech data and text processing resources available. In order to achieve this goal, we recorded a relatively small speech corpus and developed some linguistic processing tools, including a text preprocessor, a diacritic restorer and a grapheme-to-phoneme converter. In addition, we took advantage of the existing open source toolkit (HTS) for building an HMM-based speech synthesiser. The first author, who is a native speaker of Urdu, listened to several synthetic speech samples and found that some of them were not completely intelligible. However, this result is not surprising for us because this preliminary system was developed using a small speech database and a limited amount of resources for performing the text analysis in the training phase of the system. Our plan is to improve this initial version of the synthesiser by using a larger speech corpus and improving the linguistic processor, for example by incorporating a syllabification system. Also, we are going to conduct formal perceptual evaluations of speech quality and we are developing a test set of at least 1000 words with phonetic transcriptions for evaluation of the grapheme-to-phoneme converter.

7. ACKNOWLEDGEMENTS

This research is supported by the Science Foundation Ireland (Grant 12/CE/I2267) as part of CNGL (*www.cngl.ie*) at University College Dublin. The opinions, findings and conclusions, recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Science Foundation Ireland.

8. REFERENCES

[1] Black, A., Zen, H., and Tokuda, K., "Statistical parametric speech synthesis," *Proc. of ICASSP*, pp. 1229-1232, 2007.

[2] Hussain, S., "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, ISBN 99946-57-69-0, 2005.

[3] Hussain, S., "Letter-to-sound conversion for Urdu text-to-speech system," *In the Proceedings of Workshop on Computational Approaches to Arabic Script-based Languages*, Geneva, 2004.

[4] Raza, A.A., Hussain, S., Sarfraz, H., Ullah, I. and Sarfraz, Z., "Design and development of phonetically rich Urdu Speech Corpus," *In the Proceedings of O-COCOSDA*, Urunqi, China, 2009.

[5] "HMM-based speech synthesis system version 2.2", http://hts.sp.nitech.ac.jp, 2011.

[6] Rabiner, L. R., "A tutorial on hidden Markov models and selected applications," *in Speech Recognition Proc. of IEEE*, Vol. 77, pp. 257-286, 1989.

[7] Young, S., Evermann, G., Gales, M., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P., "The HTK book version 3.4", *http://htk.eng.cam.ac.uk*, 2006.

[8] Saleem, A. M., Kabir, H., Riaz, M. K., Rafique, M. M., Khalid, N. and Shahid, S. R., "Urdu consonantal and vocalic sounds," *CRULP Annual Student Report*, 2002.

[9] Raza, A. and Hussain, S., "Automatic diacritization for Urdu," *Proceedings of the Conference on Language and Technology*, 2010.

[10] Durrani, N. and Hussain, S., "Urdu word segmentation," *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 2010.

[11] Kawahara, H., Masuda-Katsuse, I. and Cheveignè, A., "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f_0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, Vol. 27, pp. 187-207, 1999.

[12] Talkin, D., "A robust algorithm for pitch tracking (RAPT)," *in Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds. 1995, pp. 495–518, Elsevier Science.

[13] "ESPS Programs Version 5.3," Entropic Research Laboratories Inc., 1998.