# A Common Attribute based Unified HTS framework for Speech Synthesis in Indian Languages

B. Ramani[1], S.Lilly Christina [1], G Anushiya Rachel [1], V Sherlin Solomi[1],
Mahesh Kumar Nandwana[2], Anusha Prakash[2], Aswin Shanmugam S [2], Raghava Krishnan[2]
S P Kishore[3], K Samudravijaya[4]
P Vijayalakshmi[1], T Nagarajan [1] and Hema A Murthy[2]

[1]Speech Lab, SSN College of Engineering, Chennai, India
[2] Department of Computer Science and Engineering, IIT Madras, India
[3] Speech and Vision Lab, IIIT Hyderabad, India
[4] TCS, TIFR Bombay, India Email: hema@cse.iitm.ac.in

## Abstract

State-of-the art approaches to speech synthesis are unit selection based concatenative speech synthesis (USS) and hidden Markov model based Text to speech synthesis (HTS). The former is based on waveform concatenation of subword units, while the latter is based on generation of an optimal parameter sequence from subword HMMs. The quality of an HMM based synthesiser in the HTS framework, crucially depends on an accurate description of the phoneset, and accurate description of the question set for clustering of the phones. Given the number of Indian languages, building a HTS system for every language is time consuming. Exploiting the properties of Indian languages, a uniform HMM framework for building speech synthesisers is proposed. Apart from the speech and text data used, the tasks involved in building a synthesis system can be made language-independent. A language-independent common phone set is first derived. Similar articulatory descriptions also hold for sounds that are similar. The common phoneset and common question set are used to build HTS based systems for six Indian languages, namely, Hindi, Marathi, Bengali, Tamil, Telugu and Malayalam. Mean opinion score (MOS) is used to evaluate the system. An average MOS of 3.0 for naturalness and 3.4 for intelligibility is obtained for all languages.

## 1. Introduction

A successful technique for speech synthesis is the unit selection-based concatenative synthesis (USS). This system selects and concatenates pre-recorded speech units in the database such that the target and concatenation costs are minimised [1]. In order to obtain high-quality synthetic speech, the size of the database required is large, to ensure that sufficient examples for each unit in every possible context is available. This results in a large footprint for USS systems.

A recent approach to speech synthesis is statistical parametric synthesis. This method involves the generation of context-dependent HMMs which are concatenated to form a sentence HMM, corresponding to the input text provided. Unlike the USS approach, the prosodic characteristics of the voice can be modified by simply varying the HMM parameters [2],[3] thereby reducing the requirement for large amount of data requirement.

A HMM-based speech synthesis system requires the following: (a) text data in a language, (b) speech data corresponding to the text, (c) time-aligned phonetic transcription, (d) context-specific features for phones if they exist (e) a question set for tying phone models. (a) and (b) are language dependent, while the rest of the modules can be made language-independent in the Indian language context.

In India most languages can be classified as Aryan and Dravidian or a mix of both. In the current work six languages are chosen Hindi, Marathi, Bengali, Tamil, Malayalam and Telugu. Indian languages have several phonetic similarities among them [4], which suggests the possibility of a compact, common phone set for all the languages. Deriving time-aligned phonetic transcription is a tedious task. The acoustic similarity among the same phones of different languages leads to a set of common, context-independent set of acoustic models that can be used for segmenting the speech signal into phonemes. Further, a common question set can also be derived that can be used for clustering in the HTS framework. This is the primary motivation for this paper. This work is motivated by the efforts of [5] in the context of building synthesis for all the languages of the world. In this work, the focus is restricted to build systems for Indian languages which number about 1652 at the time of this writing [6]. The ultimate goal is to build a generic text-to-speech system for Indian languages which can be adapted for new languages using a small amount of adaptation data.

The paper is organised as follows: Section 2 describes the speech corpora. In order to provide the appropriate context, in Section 3 the HMM based speech synthesiser is described with particular emphasis on different modules. Section 4 discusses how a common phone set, common acoustic models, and a common question set are obtained. Section 5 describes the Indian language synthesiser. Section 6 gives the performance analysis and Section 7 concludes the paper.

## 2. Speech Corpora

In the work presented in [7], speech data is collected for six of the Indian languages, namely, Tamil, Malayalam, Telugu, Hindi, Marathi and Bengali for building an USS based system. 12 hrs of speech data is collected from a female speaker (voice talent) for each of the languages separately, in a studio environment at 16KHz, 16bits/sample. The data consists of sentences from short stories, novels, science, sports, and news bulletins.

## 3. HMM-Based Speech Synthesiser

HMM-based speech synthesis consists of a training and synthesis phase. In the training phase, spectral parameters, namely, Mel generalised cepstral coefficients (mgc) and their dynamic features, the excitation parameters, namely, the log fundamental frequency (lf0) and its dynamic features, are extracted from the speech data. Using these features and the time-aligned phonetic transcriptions, context-independent monophone HMMs are trained [2]. The basic subword unit considered for the HMM-based system is the context-dependent pentaphone. For Indian language synthesis too, the pentaphone is considered as the basic subword. The UTF-8 text is converted to a sequence of pentaphones. As in conventional HTS, the context-dependent models are initialised with a set of context-independent monophone HMMs. A sequence of steps based on the common question set, is used for state-tying, which results in tree based clustering of states[8].

In the synthesis phase, again as in conventional HTS, context-dependent label files are generated for the given text and the required context-dependent HMMs are concatenated to obtain the sentence HMM. Spectral and excitation parameters are generated for the sentence and a speech waveform is synthesised. This process is illustrated in Fig. 1 [2][1]. As mentioned in Section 1, except for the text and speech data, which are language-dependent, the rest of the modules can be made language-independent by preparing a common phone set and common question set. The following section describes the common attributes across different Indian languages that can be shared to prepare common phone set, acoustic models, and question set.

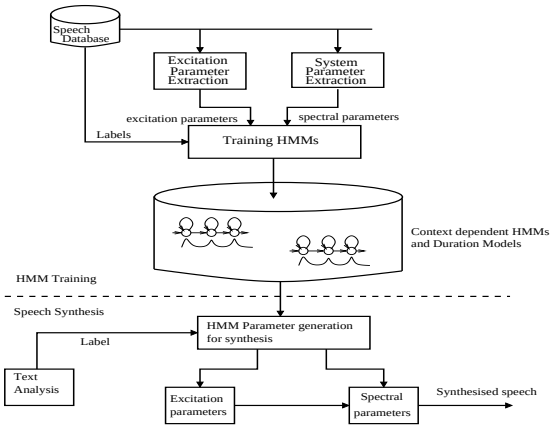


Figure 1: Overview of HMM-Based Speech Synthesis

[1]This figure has been redrawn from [2]

## 4. Sharing Common Attributes

### 4.1. Phone Set

A list of phones in each of the six languages considered in this paper, namely, Tamil, Malayalm, Telugu, Hindi, Marathi and Bengali, is shown in Figs. 2 and 3. A tag **v** is associated with some sounds for Tamil to denote the voiced counterparts. It is observed that there are 25 phones that are common to all six languages. Each language has 10 to 12 vowels, out of which 8 are common to all. Most consonants among different languages are observed to be phonetically similar. This is also confirmed by the studies of Prahalad et al. in [9]. 33 consonants are common to all languages, except Tamil, which has only 26 consonants as against 33 or 34 in others. For example, the short and long vowels for /a/, /i/ and /u/ are present in all the six languages. While Tamil, Malayalam and Telugu also have long vowels for /e/ and /o/, the Aryan languages have long vowels for /a/, /i/ and /u/ only. There are some sounds that are unique to some languages. The retroflex /zh/ is present only in Tamil and Malayalam. Tamil does not have any aspirated consonants. The palatals /c/, /ch/, /j/, /jh/ are all affricates in Indian languages. Most of the questions for Hindi were first derived from [10]. Given the common phoneset, it was observed that a Question Set of 53 questions is sufficient to cover the common phone set. Additionally, rules are included for specific languages. Altogether a set of 60 questions are prepared. The phones are grouped and mapped to labels that are closest based on the International Phonetic Alphabet (IPA). The IPA labels are modified to exclude special characters for convenience. The IPA labels map directly to the graphemes of each language (Figure 2 and 3) [2]. We now list the rationale for the development of the common phone set:

1. Rules for mapping similar sounds across languages: This common phoneset is a standard set of labels (in Roman script) for speech sounds commonly used in Indian languages. This document lists the label set for 13 languages (currently being processed by ASR/TTS consortia of TDIL, DIT, Government of India). These labels are used for computer processing of spoken Indian languages.
   (a) Similar sounds in different languages are given a single label.
   (b) The IPA symbol refers to an exemplar (Hindi/Tamil/other) language.
   (c) This is not an IPA chart of sounds of Indian languages.
   (d) The label set is designed such that the native script is largely recoverable from the transliteration.

   A label may consist of a sequence of alphanumeric characters of the Roman alphabet; they will not contain any special character such as quote, hyphen etc. All labels are in lower case even though

[2]This is a partial set. Mapping is available for all 13 languages from the authors

| Label | IPA | Hindi | Marathi | Bengali | | Tamil | Malayalam | Telugu |
|---|---|---|---|---|---|---|---|---|
| | | | | P | G | | | |
| a | /a/ | अ | अ | - | - | அ | അ | అ |
| ax | /ɔ/ | - | ऑ | অ | অ | - | - | - |
| aa | /a: / | आ | आ | আ | আ | ஆ | ആ | ఆ |
| axx | /ɔ / | - | - | - | - | - | - | - |
| i | /ɪ /,/i/ | इ | इ | ই,ি | ই | இ | ഇ | ఇ |
| ii | /i: / | ई | ई | - | ঈ | ீ | ഈ | ఈ |
| u | /u/,/ʊ / | उ | उ | উ,ু | উ | உ | ഉ | ఉ |
| eu | /ɯ / | - | - | - | - | உ | ◌ | - |
| uu | /u: / | ऊ | ऊ | - | ঊ | ூ | ഊ | ఊ |
| rq | - | ऋ,ृ | ऋ,ृ | ঋ,ৃ | ঋ,ৃ | - | ഋ | ఋ,ృ |
| e | /e/ | - | - | এ | এ | எ | എ | ఎ |
| ee | /e: / | ए | ए,ऐ | - | - | ஏ | ഏ | ఏ |
| ei | /ɛ: / | ऐ | - | - | - | - | - | - |
| ai | /aI/ | - | ऐ | - | - | ஐ | ഐ | ఐ |
| oi | /oj/ | - | - | ঐ | ঐ | - | - | - |
| o | /o/ | ओ | ओ,ओ | ও | ও | ஒ | ഒ | ఒ |
| oo | /o: / | - | - | - | - | ஓ | ഓ | ఓ |
| ae | /ae/ | - | ऍ | - | অ্যা | - | - | - |
| au | /a ʊ / | - | औ | - | - | ஔ | ഔ | ఔ |
| ou | /oʊ / | औ | - | ঔ | ঔ | - | - | - |
| k | /k/ | क | क | ক | ক | க | ക | క |
| kh | /kʰ / | ख | ख | খ | খ | - | ഖ | ఖ |
| g | /g/ | ग | ग | গ | গ | ஸ்ர | ഗ | గ |
| gh | /gʱ / | घ | घ | ঘ | ঘ | - | ഘ | ఘ |
| ng | /ŋ / | ङ | ङ | ঙ,ং | ঙ | ங | ങ | ఙ |
| c | /ʧ/ | च | च | চ | চ | ச | ച | చ |
| ch | /ʧʰ / | छ | छ | ছ | ছ | - | ഛ | ఛ |
| cx | /ʧ_̱ / | - | च | - | - | - | - | - |
| j | /dʒ/ | ज | ज | জ,য | জ | ஜ | ജ | జ |
| jh | /dʒʱ / | झ | झ | ঝ | ঝ | - | ഝ | ఝ |
| jx | /dʒ_̱ / | - | ज | - | - | - | - | - |
| nj | /ɲ/ | - | ञ | - | - | ஞ | ഞ | ఞ |
| tx | /ʈ/ | ट | ट | ট | ট | ட | ട | ట |
| txh | /ʈʰ / | ठ | ठ | ঠ | ঠ | - | ഠ | ఠ |
| dx | /ɖ / | ड | ड | ড | ড | ட்ட | ഡ | డ |
| dxh | /ɖʱ / | ढ | ढ | ঢ | ঢ | - | ഢ | ఢ |
| nx | /ɳ / | ण | ण | ণ | ণ | ண | ണ | ణ |
| t | /t/ | त | त | ত,ৎ | ত,ৎ | த | ത | త |
| th | /tʰ / | थ | थ | থ | থ | - | ഥ | థ |
| d | /d/ | द | द | দ | দ | த்த | ദ | ద |
| dh | /dʱ / | ध | ध | ধ | ধ | - | ധ | ధ |

Figure 2: Common Phone Set

| Label | IPA | Hindi | Marathi | Bengali | | Tamil | Malayalam | Telugu |
|---|---|---|---|---|---|---|---|---|
| | | | | P | G | | | |
| n | /n/ | न,ऩ | न,ऩ | ন | ন | ந,ன | ന | న |
| nd | | - | - | - | - | ந்த | - | - |
| p | /p / | प | प | প | প | ப | പ | ప |
| ph | /pʰ / | फ | फ | ফ | ফ | - | ഫ | ఫ |
| b | /b/ | ब | ब | ব | ব | ப்ப | ബ | బ |
| bh | /bʱ / | भ | भ | ভ | ভ | - | ഭ | భ |
| m | /m/ | म | म | ম | ম | ம | മ | మ |
| y | /j/ | य,य़ | य,य़ | য় | য | ய | യ | య |
| r | /r/ | र,ऱ | र,ऱ | র | র | ர | ര | ర |
| l | /l/ | ल | ल | ল | ল | ல | ല | ల |
| lx | /l_̱/ | - | ळ,ळ | - | - | ள | ള | ళ |
| w | /ʋ / | व | व | - | ওয় | வ | വ | వ |
| sh | /ʃ / | श | श | শ,ষ | শ | - | ശ | శ |
| sx | /ʂ / | ष | ष | - | ষ | ஷ | ഷ | ష |
| s | /s / | स | स | স | স | ஸ | സ | స |
| h | /ɦ / | ह | ह | হ | হ | ஹ | ഹ | హ |
| kq | /q/ | क़ | क़ | - | - | - | - | - |
| khq | /x/ | ख़ | ख़ | - | - | - | - | - |
| gq | /ɣ / | ग़ | ग़ | - | - | - | - | - |
| z | /z / | ज़ | ज़ | ज़ | - | - | - | - |
| jhq | /ʒ / | झ़ | झ़ | - | - | - | - | - |
| dxq | /ɽ/ | ड़ | ड़ | ড় | ড় | - | ऽ | - |
| dxhq | /ɽʱ / | ढ़ | ढ़ | ঢ় | ঢ় | - | - | - |
| dhq | | - | - | - | - | - | - | - |
| f | /f/ | फ़ | फ़ | - | ফ় | ◌ॅ◌ | - | - |
| bq | | - | - | - | - | - | - | - |
| yq | | - | - | - | য় | - | - | - |
| nq | | - | - | - | - | - | ന | - |
| rx | /ɾ / | - | - | - | - | ற | റ | ఱ |
| sq | | - | - | - | - | - | - | - |
| zh | /ɻ / | - | - | - | - | ழ | ഴ | ళ |
| nxh | /ɳ_̱ʰ/ | - | ण्ह | - | - | - | - | - |
| nh | /nʰ/ | - | न्ह | - | - | - | - | - |
| mh | /mʰ/ | - | म्ह | - | - | - | - | - |
| rh | /ɾʰ/ | - | ऱ्ह | - | - | - | - | - |
| lh | /lʰ/ | - | ल्ह | - | - | - | - | - |
| wh | /wʰ/ | - | व्ह | - | - | - | - | - |
| q | | ◌ं | ◌ं | - | ◌ং | - | ◌ം | ◌ం |
| hq | | ◌ः | ◌ः | - | ◌ঃ | - | ◌ഃ | ◌ః |
| mq | | ◌ँ | ◌ँ | ◌ঁ | ◌ঁ | - | - | - |

Figure 3: Common Phone Set (contd.)

the labels are case insensitive. Since the number of speech sounds are larger than the Roman alphabet, a system of suffixes as well as letter combinations are used for labels.

2. Notes on suffixes:

   (a) Aspiration: Use suffix h to denote aspiration: k (क) versus kh (ख).

   (b) Retroflex consonants: Use suffix x to denote retroflex place of articulation: t (त) versus tx (ट).

   (c) Nukta/bindu: Use suffix q to denote a nukta/bindu: dx (ड) versus dxq (ड़). Nukta (a dot below the glyph) may denote a flap/tap or a fricative variant of the consonant. Bindu (a dot above a [vowel] glyph) denotes a nasal after the vowel; the place of articulation of the nasal will be the same as that of the following consonant. If there is no consonant after the bindu, the vowel is nasalized.

   (d) Nasalized vowel: Use suffix n to denote nasalization of a vowel: k a h aa (कहा) versus k a h aan (कहाँ).

   (e) Geminated sounds: The label for a geminated consonant is the label of the corresponding single consonant with the first letter of the label repeated. Example: p a k aa (पका) versus p a kk aa (पक्का) in Hindi; a dd aa (अद्दा) in Hindi; a ll a m (అల్లమ్) (ginger in Telugu).

   (f) Other special cases: Use suffix x to denote

certain special cases: reduced vowel (axx) in various languages; "a" of Bangla; apical affricates of Marathi; special r of Dravidian languages etc.

(g) Priority of suffixes: Some symbols may have multiple suffixes. In such cases the following is the priority (in decreasing order): x h q n

3. Notes on Matras, Diphthongs and Halant:

(a) The label for a vowel matra is the same as that of the vowel.

(b) The label of a diphthong is generated as a concatenation of the labels of the corresponding vowel. The exceptions to this rule are "ae", "ea" and "eu"; these are monophthongs.

(c) The halant in Indian scripts denotes the absence of the implicit "a" in Indian consonant characters. It is not a sound and hence there is no label for halant. The morphological analyser of the language deletes the implicit "a" when a halant is present in the script.

(d) Punctuation marks: The 'transliteration' module will retain the punctuation marks (exception: '|' and "||" will be replaced by fullstops); these are useful for prosody generation. The morphological analyser will remove the punctuation marks while generating the word/phone level transcription.

4. Language specific notes: North-eastern languages have sounds (and labels) specific to a subset of the languages. Wherever required, a set of additional phonemes are defined.

Ideally, once the phones are generalised, context-independent models of acoustically similar phones across languages, can be combined. A compact set of acoustic models can be obtained to derive the time aligned phonetic transcriptions for the speech data. As this is the first attempt at a common phoneset for Indian languages, individual phone models are built for each of the languages using the aligned data.

### 4.2. Question Set

A question set is the primary requirement for tree-based clustering in an HMM-based speech synthesis system. A decision tree, that is similar to a binary tree with a yes/no question at each node is used. Relevant linguistic and phonetic classifications are included which enable accurate acoustic realisation of a phone. The question at each node of the tree is chosen such that there is a gain in the likelihood. Depending upon the answers, the phonemes for every language are split into categories, which are then tied. In the common question set, 60 common questions are formulated. Given that a common phoneset has been defined, a common question set was prepared for the languages. This set is a super set of questions across all Indian languages. This common question set that has been tested for 13 Indian languages. The number of questions in the common question set is fixed regardless of the language, since irrelevant entries in the question set are ignored while clustering [8].

## 5. Indian Language Synthesiser

### 5.1. Data Preparation

The wave files and the corresponding label files are required for building the HTS system. The common phone set for all the six languages are derived as described in Section 4.1. Common acoustic models, five minutes of speech data (phonetically balanced) for the language Tamil is considered as a representative for Telugu and Malayalam. For Aryan languages Hindi is chosen as the starting point for Marathi and Bengali. To generate unique phoneme models for the rest of the languages, few sentences are chosen in each language. Time-aligned phonetic transcriptions are obtained for this data by segmenting manually at the phoneme level using visual representations such as waveforms and the corresponding spectrograms. Monophone HMMs for all the phonemes are generated using the label files obtained. With these models, forced Viterbi alignment is performed iteratively to segment the rest of the data. This work is distinctly different from polyglot synthesis as in [11, 12] in that no attempt is made to generate common phones across different languages using the monphones from multiple languages. It is an attempt like the global phone project as in [13] to quickly build speech recognisers for various languages. This is also unlike the effort in [14], where speakers are clustered to produce a monolingual synthesiser for a new language with little adaptation data.

The effort in this paper is to primarily address the non-availability of annotated data for Indian languages. Further, there are at best only small vocabulary isolated word/isolated phrase recognition systems. Therefore, to obtain good initial monophone HMMs, a small amount of data must be manually transcribed. To reduce the effort required in manual transcription, two languages Hindi (Aryan) and Tamil (Dravidian) are first chosen, for which about 5 mins of data is manually transcribed. For languages that have additional phones, a few sentences from the given language are transcribed. This data is used to initialise the monophone HMMs. These monophone HMMs are used to force-align all the data of the appropriate group the language belongs. The HMM models are iteratively re-estimated using embedded re-estimation. Ultimately a set of language dependent HMMs are produced.

Summarising:

1. Time-aligned phonetic transcriptions are derived, for 5 mins of Tamil/Hindi speech data (phonetically balanced) manually and few sentences from each language to include unique phonemes in each language, using visual represenations such as waveforms and the corresponding spectrograms.

2. Using this data, context-independent phonemes models are trained (isolated-style training)

3. Using these models and the phonetic transcriptions (using the common phone set), the entire speech data is segmented using forced-Viterbi alignment.

4. Using the newly derived time-aligned phonetic transcription (phone-level label files), new context-independent phoneme models are trained.

5. Steps 3 and 4 are repeated $N$ times ($N = 5$, here).

6. After $N$ iterations, the HMMs are used to segment the entire speech data, again. These boundaries are considered as final boundaries.

### 5.2. Experimental Setup

Developing an HMM-based speech synthesis systems involves a training and synthesis phase. The training phase primarily requires the utterance structures, that are derived from Festival[15]. A 105 dimensional feature vector consisting of Mel-generalised cepstral (mgc) coefficients (35) their delta (35), and acceleration coefficients (35), 3 dimensional excitation features, that is, log F0 and its dynamic features are extracted from the speech files. In addition to this, the utterances are used to obtain contextual features (53 features), namely, number and position of words, syllables and phonemes, phrase breaks, stress of the subword units, which are in turn used to generate the context-dependent label files. From the parameters extracted, four-stream monophone models with five states and a single mixture component per state are generated for all 39 phonemes in the database. Five-stream duration models, with a single state and a single mixture component per state, are also generated for each phoneme. Using the common question set, tree-based clustering is carried out and context-dependent models are trained.

## 6. Performance Evaluation

To give perspective to the HTS based system, the HMM based speech synthesis system is compared with the syllable-based USS systems developed in [7]. Since Indian languages are syllable-timed and syllable is the fundamental production unit [16, 17], it is shown in [7, 16, 18, 19] that syllable-based synthesisers can be built for Indian languages with a minimal question set in the festival [15] framework. The festival based speech synthesis systems can be accessed at `http://www.iitm.ac.in/donlab/festival/`. These systems have been developed by a consortium of 12 Institutions across India and represents a set of reasonable quality text-to-speech synthesis systems developed for Indian languages. The UTF-8 encoding of the graphemes that correspond to the syllables of a language are directly used to build the systems. Part-of-speech taggers, tones and break indices markers are not readily available for all the Indian languages. Therefore, they are not used in building the synthesis systems. Similar to cluster unit based concatenative synthesis, cluster units are made from a syllable inventory. A semiautomatic algorithm [20] is used to build an inventory of syllables. As festival is primarily phoneme-centric, festival had to be modified to use syllable as a fundamental unit. A set of hand-crafted rules were developed to cluster at the syllable-level. For example, the number of units in the leaf is reduced to 20, syllables are clustered based on their position in the word, optimal coupling is turned off, etc. For details, refer to the [7]. The number of frequently used syllables is no more than 300 [21], but the syllable distribution has long tails. Fallback units to the akshara (CV) and phonemes are provided. These units are obtained by force-alignment at the syllable-level.

### 6.1. Building HTS based systems

Initially HMM-based speech synthesis systems were built for only Tamil and Hindi using the common phone set and common question set. Extensive experiments were performed to arrive at an optimal amount of data required for each of the languages. Starting from one hour data, increasing in steps of an hour it was observed that the system performance did not improve significantly beyond 5 hours of data. This evaluation was obtained by performing informal MOS tests. Based on the study on Tamil and Hindi, the systems for all six languages are built using five hrs of data [3]. The MOS is obtained for each language using sentences that are obtained from the web. Two different MOS scores are given in the Table 1. One is based on naturalness and the other is based on intelligibility. In the Table, the MOS for Tamil and Hindi corresponds to degradation MOS. This is reported based on the advice by [22]. Word-error-rates (WER) were obtained on semantically unpredictable sentences for Tamil and Hindi. The WER is indicated by W in the Table 1. The average number of subjects across the languages is about 30. The scores in Table 1 reveal that the MOS varies between 3.5 and 3.8 in terms of intelligibility, while it varies between 3.2 and 3.5 in terms of naturalness. Further, the MOS scores for festvox-based voices built [7] for the same languages, using 12 hrs of speech data, is also presented in Table 1 along with the MOS scores of HMM-based systems. This reiterates that HMM-based systems outperform in terms of MOS for intelligibility, while the USS system yields better results in terms of naturalness for all languages. The salient point of the work presented here is that: given the common Question Set, the only effort required to build a speech synthesis for a new language is mapping of the language's graphemes to the common phone set.

The website for the HTS system for the six languages is `http://www.iitm.ac.in/donlab/hts/`. Most languages support UTF-8.

## 7. Conclusion

HMM-based speech synthesis systems are built for six Indian languages, namely, Tamil, Malayalam, Telugu, Hindi, Marathi and Bengali. Owing to the phonetic similarities among Indian languages, a common phone set and a common question set are derived, thereby simplifying the task of building TTSes for Indian languages. Common acoustic models are built to hasten segmentation process. The MOS obtained for the six languages is slightly less than 3.0 in terms of naturalness, while it is higher than 3.0 in terms of intelligibility. As the phone set and question set are now generalised, a multilingual system can be developed by using adaptation techniques. Currently using the idea of the common phoneset and question set, another set of seven Indian languages TTSes in both the festival and HTS framework are at different stages of development. The success of the synthesiser does pave the way for polyglot based synthesisers for Indian languages. In particular, most Indians are trilingual (English, mother tongue and Hindi). Extending the idea of the common phone set to include languages that have

[3] At the time of this writing HMM based systems have been developed for 13 languages

Table 1: MOS for Speech Synthesised Using HTS and USS for different Indian Languages. N corresponds to naturalness, I corresponds to intelligibility, W corresponds to word error rate (in %)

| Method | Tamil | | | Telugu | | Malayalam | | Marathi | | Bengali | | Hindi | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N | I | W | N | I | N | I | N | I | N | I | N | I | W |
| HTS | 2.97 | 3.72 | 6.61% | 2.94 | 3.2 | 2.82 | 2.97 | 2.57 | 3.24 | 2.9 | 3.6 | 3.0 | 3.77 | 3.28% |
| USS | 3.23 | 3.49 | 7.52% | 3.01 | 2.65 | 3.33 | 4.1 | 3.84 | 3.58 | 3.56 | 3.59 | 3.597 | 3.602 | 7.02% |

origins in Sino Tibetan and Austric, it should be possible to have the same voice speaking any Indian language.

## Acknowledgment

The authors would like to thank the Department of Information Technology, Ministry of Communication and Technology, Government of India, for funding the project,"Development of Text-to-Speech synthesis for Indian Languages Phase II", Ref. no. 11(7)/2011-HCC(TDIL). The authors would like to thank G R Kasthuri, Asha Talambedu, Jeena Prakash and Lakshmi Priya for performing MOS tests.

## 8. References

[1] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, 1996, pp. 373–376.

[2] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, pp. 1039–1064, November 2009.

[3] K. Tokuda, H. Zen, and A. W. Black, "An hmm-based speech synthesis system applied to english," in *Speech Synthesis*, 2002, pp. 227–230.

[4] A. K. Singh, "A computational phonetic model for Indian language scripts," in *In Constraints on Spelling Changes: Fifth International Workshop on Writing Systems*, 2006.

[5] "Simple 4 all," http://www.simple4all.org, Centre for Speech Technology Research, Edinburgh, 2011.

[6] wikipedia, "Languages of India," http://en.wikipedia.org/wiki/Languages_of_India, 2013.

[7] H. A. Murthy and et al., "Syllable-based speech synthesis for Indian languages and their integration with screen readers," *Speech Communication*, 2012 (submitted).

[8] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*, Cambridge University Engineering Department, 2002.

[9] L. Prahalad, K. Prahalad, and G. L. Madhavi, "A simple approach for building transliteration editors for Indian languages," in *Journal of Zhejiang University Science*, vol. 6A, no. 11, 2005, pp. 1354–1361.

[10] P. Eswar, "A rule based approach for spotting characters from continuous speech in Indian languages," PhD Dissertation, Indian Institute of Technology, Department of Computer Science and Eng., Madras, India, 1991.

[11] J. Latorre, K. Iwano, and S. Furui, "Polyglot synthesis using a mixture of monolingual corpora," in *icassp*, 2005, pp. 1–4.

[12] ——, "New approach to polyglot synthesis: how to speak any language with anyone's voice," in *Proceedings of the ISCA Tutorial Research Workshop on Multilingual Speech and Language Processing*, April 2006.

[13] T. Schultz, "Global phone project," http://cs.cmu.edu/ tanja/GlobalPhone/index.html, 2000.

[14] A. Black and T. Schultz, "Speaker clustering and multilingual synthesis," in *Proceedings of the ISCA Tutorial and Research Workshop on Multilingual Speech and Language Processing*, April 2006.

[15] A. Black, P. Taylor, and R. Caley, "The Festival speech synthesis system," `http://festvox.org/festival/`, 1998.

[16] S. Arora, K. K. Arora, and S.S.Agrawal, "Using syllable as a major unit for developing an efficient concatenative hindi speech synthesiser," in *Proc. of the Int. conf. SPECOM*, 2005, pp. 675–679.

[17] J. Cholin and W. J. M. Levelt, "Effects of syllable preparation and syllable frequency in speech production: Further evidence for syllabic units at a post-lexical level," *Language and Cognitive Processes*, no. 24, pp. 662–682, 2009.

[18] M. N. Rao, S. Thomas, T. Nagarajan, and H. A. Murthy, "Text-to-speech synthesis using syllable-like units," in *National Conference on Communication*, 2005, pp. 227–280.

[19] S. Thomas, M. N. Rao, H. A. Murthy, and C. S. Ramalingam, "Natural sounding speech based on syllable-like units," in *EUSIPCO, Florence, Italy*, 2006.

[20] H. A. Murthy and B. Yegnanarayana, "Group delay functions and its application to speech processing," *Sadhana*, vol. 36, no. 5, pp. 745–782, November 2011.

[21] V. K. Prasad, "Segmentation and recognition of continuous speech," PhD Dissertation, Indian Institute of Technology, Department of Computer Science and Eng., Madras, India, 2002.

[22] S. King, "Degradation MOS and word error rate for text to speech synthesis systems," private Communication.