Building Hindi and Telugu Voices using Festvox

S.P. Kishore, Rajeev Sangal and M. Srinivas
Language Technologies Research Center
International Institute of Information Technology Hyderabad
Email: {kishore, sangal}@iiit.net, srinu@gdit.iiit.net

Abstract

In this paper, we discuss the development of Hindi and Telugu voices using Festvox. Relevant details to implement a text-to-speech system for Indian languages using Festvox are given. We also present an application on handheld devices, called "talking tourist-aid" which will assist the tourists in interacting with local people in their native language.

1 Approaches for Text to Speech Conversion

The function of Text-To-Speech (TTS) system is to convert the given text to a spoken waveform. This conversion involves text processing and speech generation processes. These processes have connections to linguistic theory, models of speech production, and acoustic-phonetic characterization of language [Klatt, 1987] [Yegnanarayana *et al.*, 1994].

Approaches to build a TTS system can be divided into three broad categories: 1) articulatory-model based approach, 2) parameter-based approach and 3) strategies for concatenating the stored speech segments. In articulatory-model based synthesis, simplified models of the articulators or models of the observed shape of the vocal tract are devised and a set of rules are specified to control the position of the articulators. Such TTS systems are found to generate natural sounding speech but have the difficulty in acquiring sufficient data on the motion of the articulators during speech production. In parameter-based approach, parameters such as formant frequencies are manipulated according to heuristic rules formed by observing the spoken data. These rules incorporate the prosodic details such as intonation and duration patterns and phonetic details including the complexities such as nasalization of vowels. Several hundred precisely crafted rules are needed to control a formant synthesizer [Klatt, 1987].

The third approach is to concatenate stored speech segments. These speech segments (also referred to as units) cannot be words due to prosodical variations in the isolated words and the words spoken in a sentence. In a sentence, words are as short as half their duration when spoken in isolation. At the

same time, concatenation of strings of phoneme-sized units have failed because of articulatory effects between adjacent phoneme that cause substantial changes to the acoustic manifestation of a phoneme depending on context. Thus sub-word units such as syllables and diphones in which coarticulation between adjacent phonemes is preserved are considered as satisfactory units.

Diphone, an acoustic chunk from the middle of one phoneme to the middle of another phoneme is widely used for English, as TTS systems seem to function with a small inventory of about 1000 diphones. These TTS systems modify the duration and fundamental frequency contours of the prosodically neutral diphone according to the required context. An alternative is to store multiple realizations of each unit with differing prosody [Klatt, 1987]. Current TTS systems widely employ this technique of storing multiple realizations of each unit with differing prosody [Hunt and Black, 1996] [Kenney Ng, 1998]. These TTS systems are shown to generate more natural speech than the conventional approaches. A suitable term for these approaches is *data-driven synthesis*. Typically, there is a large database of speech with variable number of acoustic manifestations of each unit. During synthesis, a particular manifestation of a unit is selected depending on how well it matches with the input specification and on how well it matches with other units in the sequence.

In this paper, we discuss the development of TTS systems for Indian languages using Festvox. It should be noted that there are efforts to develop TTS systems for Indian languages using hybrid models [Rajeshkumar, 1990], [Yegnanarayana *et al.*, 1994] and formant synthesizers [Sen and Samudravijaya, 2002]. This paper is organized as follows: Section 2 describes the phonetic nature of Indian languages. Section 3 proposes the Festvox framework to build Hindi and Telugu voices. Development of talking tourist-aid is discussed in Section 4.

2 Phonetic Nature of Indian Languages

The scripts of Indian languages have originated from the ancient Brahmi script. The basic units of writing system are *characters* which are orthographic representation of speech sounds. A character in Indian language scripts is close to syllable and can be typically of the following form: C, V, CV, CCV and CVC, where C is a consonant and V is a vowel. There are about 35 consonants and about 18 vowels in Indian languages.

An important feature of Indian language scripts is their phonetic nature. There is more or less one to one correspondence between what is written and what is spoken. The rules required to map the letters to sounds of Indian languages are almost straight forward. All Indian language scripts have common phonetic base.

3 Festvox for Building Voices

Festvox is a collection of tools and scripts that allows voices to be built in both existing and new languages. It supports data-driven synthesis algorithm known as unit selection algorithm [Black and Taylor, 1997] [Black and Lenzo, 2000a].

3.1 Building Hindi and Telugu Voices

To build a voice in a new language, the steps involved are as follows:

- Defining the phone set of the language
- Incorporation of letter-to-sound rules
- Incorporation of syllabification rules
- Assignment of stress patterns to the syllables in the word
- Generation of speech database
- Labeling the speech database
- Extraction of pitch markers and Mel-frequency cepstral coefficients
- Building the units' database by clustering algorithm

In defining the phone set for Indian languages, we have followed a lower case notation, called *Z notation* (Appendix B) to transliterate the Hindi and Telugu scripts onto the machine.

3.2 Letter to Sound Rules, Syllabification and Stress Patterns

Letter-to-sound rules are almost straight forward in Indian languages, as they are phonetic in nature. We almost speak what we write, and hence generally the necessity of a pronunciation dictionary does not arise in our case. The pronunciation for a Telugu word such as *nagaramz* (town) in terms of phones marked with syllable boundaries can be written as ((n a) 1) ((g a) 0) ((r a mz) 0). As the characters in Indian language are close to a syllable, clustering C^*VC^* can be done easily taking into account a few exceptions.

In this work, simple syllabification rules are followed. Syllable boundaries are marked at the vowel positions. If the number of consonants between two vowels is more than one, then first constant is treated as coda of the previous syllable and the rest of the consonant cluster as the onset of the next syllable. For stress assignment, the primary stress is associated with the first syllable and secondary stress with the remaining syllables in the word. The integer "1" assigned to first syllable in the word *nagaramz* indicates the primary stress associated with it. Letter to sound rules, syllabification rules and assignment of stress patterns for a new language can be implemented easily in Festvox. The architecture of Festival synthesis engine allows these rules to be written in Scheme, so that they get loaded at the runtime, essentially avoiding recompilation of the core code for every new language.

We also need to assign inherent durations to these phones, which will be useful for automatic labeling of speech database. In Appendix A, the mean and standard deviation of the durations of the phones in Telugu language obtained from the labelled data is given. This information can be used as *a priori* duration knowledge in the development of TTS systems for other Indian languages.

3.3 Hindi and Telugu Speech Databases

The quality of data-driven synthesis approaches is inherently bound to speech database from which the units are selected [Black and Lenzo, 2001]. It is important to have an optimal speech corpus balanced in terms of phonetic coverage and the diversity in the realizations of the units. In this work, speech databases are generated from a set of sentences selected from a large text corpus available in Indian languages [Bharati *et al.*, 1998].

The details of the actual algorithm are as follows: Given the text corpus and the list of syllables with frequency count, the syllable list is limited based on a threshold on frequency count. A sentence is selected from the text corpus, if it has at least one high frequency syllable, not present in the previous selected sentences. Note that this particular syllable could be available in the sentences selected further, on account of unavailability of some other syllable. So, once the selection has been done, the sentences are scanned and a few sentences are removed, if the syllables available in these sentences are also available in the remaining sentences.

3.4 Implementation of Hindi and Telugu TTS

The text selection approach mentioned in Section 3.3, ensures the coverage of high frequency syllables of a language. Using this approach, we arrived at a set of sentences in Telugu and Hindi. The selected sentences are recorded in a quiet room. The duration of Telugu speech data recorded for this purpose is around 110 minutes, while the duration of Hindi speech data is around 96 minutes. The Telugu speech corpus contains 33,417 realizations of 2,291 syllable units, and the Hindi speech corpus contains 23,179 realizations of 2,391 syllables.

These databases are labeled at the phone level with the labeler provided by Festvox. This labeler uses a dynamic time warping approach, and since accurate duration knowledge was not available for Indian language phones, the label boundaries were not accurate. These label boundaries are corrected manually using *emulabel* (www.festvox.org/emulabel). Festvox is used to extract pitch markers and Mel-cepstral coefficients, and then to build a decision tree for each unit (phone) based on questions concerning the phonemic and prosodic context of that unit.

During synthesis, for each unit (to be synthesized) the unit selection algorithm of Festvox selects an appropriate decision tree and searches for a suitable manifestation of the unit which is close to its cluster center and optimizes the cost of joining two adjacent units. Speech synthesized by the Hindi and Telugu voices for an arbitrary text was subjected to the test of native speakers and were asked to listen to synthesizer speaking a small portion of news bulletin. The output of the synthesizer was perceived to be fairly intelligible and natural.

4 Talking Tourist-aid

If we exploit the domain specific knowledge, or context in which the synthesizer is employed, the quality of the synthesized speech would be much better [Black and Lenzo, 2000b]. Applications such as talking clock, railway information systems and airway information systems have limited vocabulary

and constraints in the structure of the sentence. For example to tell the time in Telugu, the typical format of the natural spoken sentence is: "$\frac{ippudzu}{now}$ $\frac{samayamz}{time\_is}$, [period] [hour] $\frac{gamztzalaa}{hours}$ [minute] $\frac{nimishzaalu}{minutes}$". While the variables [period], [hour] and [minute] would take the possible values, the construction of the sentence is limited to the specified format. An important issue in developing voices for such applications would be the design of the prompt list that adequately covers the required domain.

One such useful application which we have developed is talking tourist-aid. This application is intended to work on a hand-held device such as Simputer (www.simputer.org). The tourist-aid would allow a tourist who knows English but does not know the local Indian language, to convey his questions or concerns to somebody who knows the local language. The interface of the application is a menu containing English queries, frequently used phrases, dictionary, etc. which the tourist can select out of, by means of a stylus. The machine would speak it out in an Indian language. The receiver can reply with gestures, simple English words (such as yes, no), etc. Some realistic situations such as seeking information for "travel" by "train, bus or aeroplane" are also incorporated. For example, if the selected question is "What is departure time for Chennai Train?", this would be spoken out in the appropriate native language, say in Telugu as "$\frac{chennai}{Chennai}$ $\frac{velzle\_railu}{train}$ $\frac{ennimztiki}{at\_what\_time}$ $\frac{bayaludezru\_tumzdi}{will\_it\_depart}$". The expected answer could be "four" or "four o'clock" providing a useful hint for the non-native tourist. Other such scenarios include 1) direction: seeking the directional information about locations such as railway station, hospital and telephone booth, 2) restaurant and 3) hotel. The output of the synthesizer for talking tourist-aid was found to be near natural in both Hindi and Telugu.

5 Conclusions and Future work

In this paper, we discussed the development of Text-To-Speech systems for Indian languages under the frame work of data-driven approach using Festvox. We generated Hindi and Telugu speech databases taking into account the high frequency syllables occurring in these languages. We developed Hindi and Telugu synthesizers using Festvox and the quality of these synthesizers was observed to be fairly intelligible and natural. We also discussed the talking tourist-aid for hand-held devices.

The future work will be focussed on improving the quality of the general purpose synthesizer. We are refining the text selection algorithm to obtain a phonologically balanced set of sentences. To incorporate more naturalness in the synthesized speech signal, we are also experimenting with syllable like units as the basic units for concatenation.

References

[Bharati *et al.*, 1998] Bharati, Akshar, Sushma Bendre, and Rajeev Sangal. Some observationson corpora of some Indian languages. In *Knowledge-Based Computer Systems,Tata McGraw-Hill*, 1998.

[Black and Lenzo, 2000a] Alan W. Black and Kevin Lenzo. Building voices in the festival speech synthesis system, www.festvox.org/festvox/index.html, 2000.

[Black and Lenzo, 2000b] Alan W. Black and Kevin Lenzo. Limited domain synthesis. In *Proceedings of Int. Conf. Spoken Language Processing*, 2000.

[Black and Lenzo, 2001] Alan W. Black and Kevin Lenzo. Optimal data selection for unit selection synthesis. In *ISCA, 4th Speech Synthesis Workshop*, 2001.

[Black and Taylor, 1997] Alan W. Black and P. Taylor. Automatically clustering similar units for unit selection in speech synthesis. In *Proceedings of EUROSPEECH'97*, pages 601–604, 1997.

[Hunt and Black, 1996] A.J. Hunt and Alan W. Black. Unit selection in a concatenative speech synthesis system for a large speech database. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 373–376, 1996.

[Kenney Ng, 1998] Kenney Ng. Survey of data-driven approaches to speech synthesis, citeseer.nj.nec.com/ng98survey.html, 1998.

[Klatt, 1987] D.H. Klatt. Review of text-to-speech conversion for english. *J. Acoust. Soc. Amer.*, pages 737–793, 1987.

[Rajeshkumar, 1990] S.R. Rajeshkumar. *Significance of Durational Knowledge for a Text-to-Speech System in an Indian Language*. MS dissertation, Indian Institute of Technology, Department of Computer Science and Engg., Madras, 1990.

[Sen and Samudravijaya, 2002] A. Sen and K. Samudravijaya. Indian accent text to speech system for web browsing. *Sadhana*, 2002.

[Yegnanarayana *et al.*, 1994] B. Yegnanarayana, S. Rajendran, V.R. Ramachandran, and A.S. Madhukumar. Significance of knowledge sources for a text-to-speech system for Indian languages. *Sadhana*, pages 147–169, 1994.

Appendix A: Duration of Telugu Phones

The duration of Telugu phones obtained in terms of mean and standard deviation in seconds.

(dhz 0.042655 0.010826)
(chh 0.158935 0.101032)
(nzz 0.050313 0.014616)
(nzzz 0.050313 0.014616)
(rz 0.031720 0.017013)
(ez 0.161353 0.104335)
(thz 0.087511 0.039344)
(gh 0.082165 0.081332)
(oo 0.161282 0.077916)
(au 0.143691 0.062236)
(kh 0.153268 0.061402)
(ph 0.177173 0.099108)
(tz 0.087119 0.042988)
(ii 0.159445 0.113648)
(uu 0.151886 0.099567)

(lz 0.053461 0.015681)
(x 0.106902 0.077511)
(b 0.123149 0.093731)
(shz 0.122901 0.059902)
(bh 0.115179 0.075954)
(sh 0.155882 0.087442)
(dz 0.050385 0.038969)
(dh 0.071263 0.034539)
(nz 0.049356 0.023208)
(v 0.081708 0.067402)
(l 0.066312 0.034801)
(ai 0.160355 0.081913)
(t 0.109699 0.063485)
(aa 0.156853 0.083281)
(g 0.078842 0.061014)
(y 0.065817 0.051982)
(o 0.132318 0.092224)
(ch 0.120280 0.059532)
(p 0.112095 0.066462)
(j 0.136147 0.097125)
(r 0.049694 0.041021)
(i 0.112849 0.104900)
(s 0.141991 0.079234)
(m 0.092399 0.064361)
(th 0.085734 0.029871)
(e 0.125599 0.100779)
(n 0.074083 0.058157)
(k 0.110862 0.059837)
(u 0.112334 0.102646)
(d 0.076272 0.068739)
(mz 0.089877 0.052955)
(a 0.091049 0.079705)
(SIL 0.452781 0.702883)

Appendix B: Z Notation

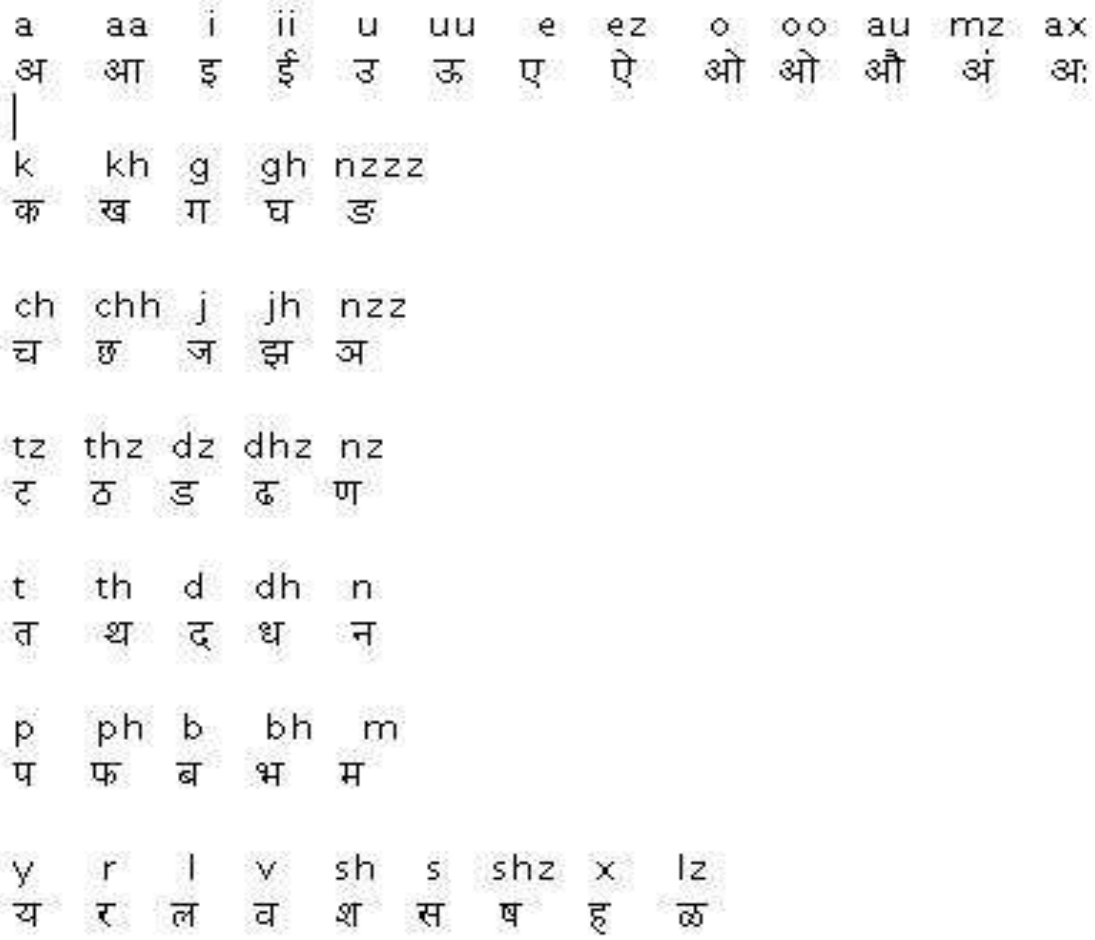

Figure 1: Aksharas in Z notation.