A Data-Driven Synthesis Approach For Indian Languages using Syllable as Basic Unit

S.P. Kishore †, Rohit Kumar ‡ and Rajeev Sangal †
† Language Technologies Research Center
International Institute of Information Technology, Hyderabad, India
‡ Punjab Engineering College, Chandigarh, India
Email: {kishore@iiit.net, rohit@pec.ac.in, sangal@iiit.net}

Abstract

This paper describes a data-driven synthesis method for Indian languages using syllables as basic units for concatenation. Unit selection algorithm of this method exploits the advantages of a prosodic matching function, which is capable of implicitly selecting a larger sequence such as words, phrases and even sentences. We show that the proposed method generates high quality speech for Indian languages.

1 Text to Speech Systems

The function of Text-To-Speech (TTS) system is to convert an arbitrary text to a spoken waveform. This generally involves two steps: 1) text processing: converting the given text to a sequence of synthesis units and 2) speech generation: generation of an acoustic wave form corresponding to each of these units in the sequence. [Klatt, 1987] [Yegnanarayana *et al.*, 1994].

Current TTS systems employing data-driven synthesis techniques are shown to generate more natural speech than the conventional approaches [Hunt and Black, 1996] [Black and Taylor, 1997] [Kenney Ng, 1998]. Data-driven synthesis techniques employ a large speech corpus containing multiple realizations of each unit with differing prosody. 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. Unit selection algorithm tries to select the longest available strings of units that match a sequence of target units.

In [Hunt and Black, 1996], the units are labeled with the available phonemic information (labels of neighboring units, position in phrase), prosodic features (energy, duration and pitch), and acoustic features such as spectral tilt. For synthesis, the prosodic features such as pitch, duration and power are predicted for target units. A particular realization of a unit is selected depending on how well it minimizes the cost function of unit distortion and discontinuity measures. Optimal set of weights are derived to minimize the cost function by weight space search or regression training techniques. The synthesis approach of [Black and Taylor, 1997] builds decision trees for the units of the same type based on questions concerning prosodic and phonemic context and its acoustic similarity with the neighboring units. During synthesis, for each target unit, the appropriate decision tree is used to find the best cluster of candidate units. A search is then made to find the best path through the candidate units that takes into account the distance of a candidate unit from its cluster center and the cost of joining two adjacent units. These approaches use phone as basic unit and either involve prediction of features for target units or use classification and regression techniques to build decision trees. In [Black and Taylor, 1997] an acoustic measure is used to split the units of same type into small clusters.

In this paper, we propose a data-driven synthesis method using syllables as basic units for concatenation. This method uses a prosodic matching function based on pitch, duration and energy for selecting an appropriate realization of a syllable. We show that the defined prosodic matching function implicitly selects a possible larger unit such as word, phrase or even sentence. The efficiency of this synthesis approach is demonstrated for Indian languages.

It should be mentioned that there are earlier works in the direction of developing TTS system for Indian languages [Rajeshkumar, 1990], [Yegnanarayana *et al.*, 1994] and [Sen and Samudravijaya, 2002]. In [Rajeshkumar, 1990] and [Yegnanarayana *et al.*, 1994], CV (Consonant-Vowel) and CCV (Consonant-Consonant-Vowel) units are used as basic units under the framework of parameter concatenation techniques. The TTS system of [Sen and Samudravijaya, 2002] is based on formant synthesis techniques. Our work focuses on developing TTS system for Indian languages under the framework of data-driven synthesis techniques.

This paper is organized as follows: Section 2 describes the features of Indian language scripts and sounds. In Section 3, the method followed to generate a speech database is described. The synthesis algorithm is discussed in Section 4. A new joining function based on prosodic parameters is defined in Section 4.1. Development of Hindi TTS and its performance evaluation is discussed in Section 5.

2 Features of Indian Language Scripts and Sounds

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.

2.1 Letter to Sound Rules

There is almost one to one correspondence between what is written and what is spoken in Indian languages. Each character in Indian language script has a correspondence to a sound of that language. For example the word *kamala* (lotus) can be easily mapped to sequence of consonant and vowel sounds */k/ /a/ /m/ /a/ /l/*, taking into account a few exceptions. These exceptions are associated with inherent vowel suppression, as is the case of */la/* in *kamala* which is mapped to */l/* alone as supposed to */l/ /a/*.

In Indian languages, a consonant character is inherently bound with the vowel sound /a/, and is almost always pronounced with this vowel. In some occurances this vowel is not pronounced, and this is referred to as *Inherent Vowel Suppression (IVS)* [Rajeshkumar, 1990]. This occurs at both word final and word middle positions. A few heuristic rules to detect IVS of a consonant character are noted below. These rules have been derived by observing a few Hindi words, and the rule set may not be a complete description of the phenomenon.

R.1 No two successive characters undergoes IVS.

R.2 The last character of the word always have its vowel suppressed unless the vowel is not /a/.

R.3 For characters in word middle position, IVS occurs if the next character in the word is not the last character or the next character has a vowel other than /a/.

2.2 Syllabification Rules

While letter to phone rules are almost straight forward in Indian languages, the syllabification rules are not trivial. There is need to come up with some rules to break the word into syllables. We have derived certain simplistic rules for syllabification i.e. rules for grouping clusters of C*VC* based on heuristic analysis of several words in Telugu and Hindi languages.

1 When nasals such as */mz/* (half pronounced /m/ sound, see Appendix A) succeed a vowel immediately, they are treated as a part of the vowel and the same syllable. For example, */mz/* in *samzskrit* (language) will be a part of syllable containing */sa/*.

2 Whenever there are three or more consonants between two consecutive vowels, the first consonant would be a part of the coda of the previous syllable while the remaining consonants would be onset of the next syllable. Applying these rules to *samzskrit*, the obtained syllable sequence would be */samzs/ /krit/*.

3 When there are exactly two consonants between two vowels, the first consonant would be part of coda of previous syllable and the second would be onset of the next syllable. For example, *dharti* (earth) would be split as it /dhar/ /ti/. Exceptions for this rule are the following cases.

   3.1 When the second consonant is a member of the set { */r/ /s/ /sh/ /shz/* }, both the consonants would be a part of onset of the next syllable. For example, *yaatra* (tour) would be split as */yaa/ /tra/*.

3 Generation of Speech database

The synthesis quality of data-driven methods is inherently bound to the speech database from which the units are selected. Design and development of a speech corpus balanced in terms of coverage and prosodic diversity of all the units is one of the major issues in such methods. In this work, our effort has been to generate a speech database which ensures coverage of high frequency syllables of the language. A syllable is called to be a high frequency syllable if its frequency (occurance) count in a given text corpus is relatively high. We have made use of large text corpus available with frequency count of the syllables in Indian languages [Bharati *et al.*, 1998]. In a sequential selection a sentence is selected from the text corpus if it contains at least one unique instance of a high frequency syllable, not present in the previous selected sentences.

This set of selected sentences are recorded in a quiet room with a noise canceling microphone using the recording facilities of a typical multimedia computer system. The speech database is labeled at the phone level and marked with syllable boundaries. For each syllable, the phonetic context such as last phone of the previous syllable, next phone of the succeeding syllable and prosodic features such as pitch, duration and energy of its own as well as of the previous and succeeding syllables are stored. The position of the syllable in the word is also stored. A similar information is stored for each phone available in the speech database.

4 Synthesis Approach

The text to be synthesized is analyzed and broken into sequence of syllables to be selected and concatenated. The proposed approach selects a syllable such that a larger sequence can also be selected, if present in the database. In cases, when there is no single instance of the required syllable, the particular syllable is manifested by concatenation of diphones. Details of the selection algorithm are as follows.

Having selected the first $k-1$ syllables, a manifestation of k^{th} syllable is selected if it has the following features.

- Phonetic context: The k^{th} syllable should have the coarticulation effect of last phone of the $k-1^{th}$ syllable
- Its position in the word is same as required.
- Prosodic features such as energy, pitch and duration match the expectations of $k-1^{th}$ syllable. The similarity match between prosodic features of units is measured by prosodic matching function as defined in Section 4.1.

If a realization of the k^{th} syllable does not posses the first two features, then the selection is based purely on prosodic matching function. A realization of the k^{th} syllable which has least value of prosodic matching function is selected for concatenation. A special case of this algorithm is the selection of first syllable. Since there is no previous syllable for first syllable, it is selected based on its succeeding syllable. An optimal pair of first and second syllables is selected jointly from the database.

4.1 Prosodic Matching Function

The prosodic matching function is based on the simple premise that a unit such as syllable selected from a speech database is perceived better if its successor (and predecessor) in synthesized speech has the same or at least similar prosodic features as that of its actual successor (and predecessor) in the speech database.

The function $m(.)$ defined below evaluates how best the k^{th} syllable matches the expectations of $k-1^{th}$ syllable. Let E^e, P^e, D^e denote the expected energy, pitch and duration of the $k-1^{th}$ syllable for its successor (i.e k^{th} syllable) in the synthesized speech. Note that these expected values are the actual values of the syllable following $k-1^{th}$ syllable in the speech database from which $k-1^{th}$ syllable is extracted. Let E^a, P^a, D^a denote the actual energy, pitch and duration of the k^{th} syllable.

$$m = \| 1 - \frac{E^a}{E^e} \| + \| 1 - \frac{D^a}{D^e} \| + \| 1 - \frac{P^a}{P^e} \| \tag{1}$$

A lesser value for the function $m(.)$ denotes a better prosodic match between k^{th} and $k-1^{th}$ syllable. The function $m(.)$ attains the possible least value i.e zero, when $E^a = E^e$, $D^a = D^e$ and $P^a = P^e$. This is possible only when k^{th} syllable (under consideration for selection) is next to $k-1^{th}$ syllable in the speech database too. By selecting such unit the matching function has implicitly selected a larger unit comprising of two syllables i.e $k-1^{th}$ and k^{th} syllables. Following the same logic in the case of selection of $k+1^{th}$ syllable, we can say that the prosodic matching function $m(.)$ selects a large sequence, if available in the database. This leads to implicit selection of complete words, phrases and even sentences when they are actually present in the database.

When the function $m(.)$ does not attain the possible least value zero, then a realization of the k^{th} syllable is selected whose value of $m(.)$ is least among the other realizations.

5 Hindi TTS system

The proposed synthesis approach is applied to develop a Hindi text to speech system. The duration of the speech data recorded for this purpose is around 96 minutes. It has 2391 high frequency syllables of Hindi language and the total number of realizations of all these syllables is 23096. The total number of realizations of 49 phone units is around 54734. The number of diphones available in this database are 1431.

To evaluate the quality of Hindi TTS, a random set of sentences were synthesized and played to the native Hindi speakers. The synthesized speech was found to be fairly intelligible and natural. A comparison in the perceptual quality was made with the sentences synthesized from Hindi TTS system built using Festvox and its selection algorithm [Black and Lenzo, 2000]. Unit selection algorithm in Festvox uses phones (which includes some part of previous phone, thus it can be treated as implicit use of diphones) as units for concatenation. The quality of the syllable-based approach was found to be better than that of Festvox. We believe that the better quality of our proposed approach is due to prosodic matching function which implicitly selects larger possible sequence and also due to use of syllable as basic unit for concatenation, a larger unit than diphone used in Festvox, thus reducing the number of joints.

6 Conclusions

The proposed approach minimizes the coarticulation effects and prosodic mismatch between two adjacent units. Selection of a unit is based on the phonetic context and the prosodic parameters of its adjacent (previous or succeeding) unit. Our hypothesis has been that the procedure of selecting a unit satisfying the local phonetic constraints and prosodic constraints through prosodic matching function which will implicitly selects a larger available sequence leads to high quality synthesis. We have observed the efficiency of this approach for Indian languages and found that the performance of this approach is better than that of other data-driven synthesis techniques adapted for Indian languages.

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

[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.

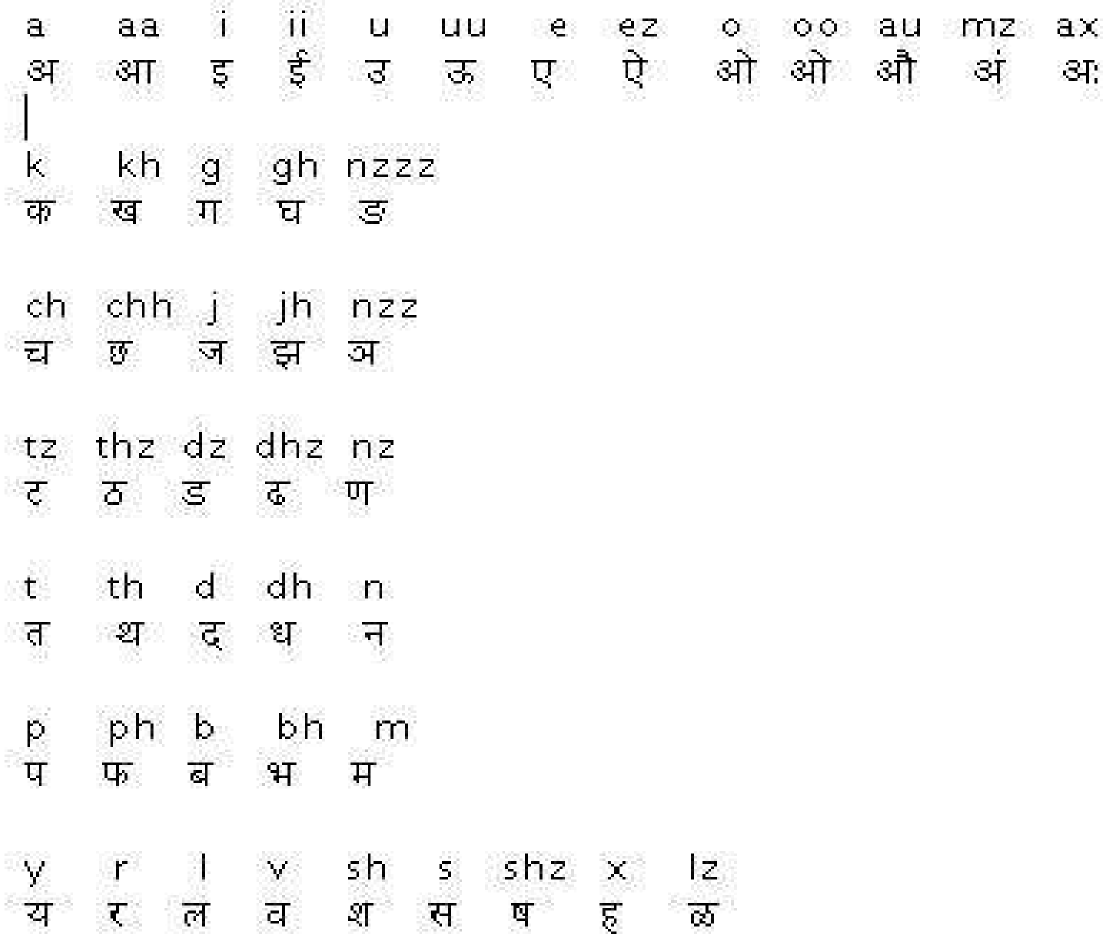
[Klatt, 1987] D.H. Klatt. Review of text-to-speech conversion for english. *J. Acoust. Soc. Amer.*, pages 737–793, 1987.

[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: Z Notation



a	aa	i	ii	u	uu	e	ez	o	oo	au	mz	ax
अ	आ	इ	ई	उ	ऊ	ए	ऐ	ओ	ओ	औ	अं	अः

|

k	kh	g	gh	nzzz
क	ख	ग	घ	ङ

ch	chh	j	jh	nzz
च	छ	ज	झ	ञ

tz	thz	dz	dhz	nz
ट	ठ	ड	ढ	ण

t	th	d	dh	n
त	थ	द	ध	न

p	ph	b	bh	m
प	फ	ब	भ	म

y	r	l	v	sh	s	shz	x	lz
य	र	ल	व	श	स	ष	ह	ळ

Figure 1: Aksharas in Z notation.