BUILDING A TAMIL VOICE USING HMM SEGMENTED LABELS

Santhosh Yuvaraja †, *Venkatesh Keri* †, *Sathish Chandra Pammi* †
Kishore Prahallad †§, *Alan W Black* §

†International Institute of Information Technology, Hyderabad, India
§Language Technologies Institute, Carnegie Mellon University, USA

ABSTRACT

In this paper, we describe the development of unit selection voice for Tamil language. We describe the build process and address the issue of speech segmentation using HMM based techniques. We report the comparison of automatically segmented labels of Sphinx-II and HTK with manually segmented labels in the context of Hindi database. Our studies show that the the segmentation accuracies of Sphinx-II and HTK are nearly same when compared to manual segmentation and the use of delta and delta-delta features may not be significant for speech segmentation.

1. INTRODUCTION

Given the increased availability of digital content in local languages, and the advent of Digital Library portal of India [1], appliances such as PCTVT for illiterate and common people [2], there is a real need and a set of real users asking for speech synthesis systems in all of the Indian languages. Following our previous work in building Hindi and Telugu voices [3], we have continued to add more Indian languages and in the process we have built a unit selection voice for Tamil. In this paper we describe the nature of Tamil scripts, letter to sound rules, syllabification rules and the development of unit selection voice for Tamil. This work is done within the FestVox voice building framework [4], which offers general tools for building unit selection synthesizers in new languages. FestVox offers a language independent method for building synthetic voices, offering mechanisms to abstractly describe phonetic and syllabic structure in the language. To build a unit selection voice in a new language, it is required to record speech from a single speaker and obtain the segmentation labels of the recorded speech. The Festvox framework supports the use of SphinxTrain [5] for training the acoustic models for the given voice and Sphinx-II [6] decoder for obtain segmentation labels using forced-alignment. A similar function could be performed using Hidden Markov Model Tool Kit (HTK) [7]. However, it is often not clear how well these tools perform the segmentation when compared to manual segmentation. It is also not clear whether the feature relevant to speech recognition hold their relevance and significance in the context of speech segmentation. In this paper we address these issues by comparing the segmentation label produced by Sphinx-II and HTK with that of manually segmented labels using Hindi speech database. Our results show that the commonly used (smoothed) delta and delta-delta features may not be significant features for speech segmentation. We also show that context-independent models may be more useful for speech segmentation in the context of speech synthesis.

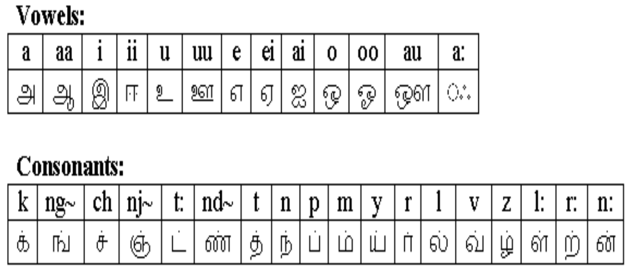
This paper is organized as follows: Section 2 describes the nature of Tamil scripts. Section 3 discusses the letter to sound rules and syllabification rules. Sections 4-5 discuss the recording of Tamil speech database and the segmentation of the recorded speech. Section 6 discusses the build process and a small-scale perceptual evaluation of the synthesized speech. Sections 7-9 discuss our experiments and results on speech segmentation.

2. NATURE OF TAMIL SCRIPTS

The basic units of the writing system in Indian languages are characters which are an orthographic representation of speech sounds. A character in Indian language scripts is close to a syllable and can be typically of the form: C^*VC^*, where C is a consonant and V is a vowel. There is fairly good correspondence between what is written and what is spoken. Typically there are about 35 consonants and 18 vowels characters. However, in Tamil there are fewer characters than many of the other Indian languages. Fig. 1 shows the vowels and consonants of Tamil used in our text to speech system. There are 13 vowels and 18 consonants characters. Some of the consonants have more than one pronunciation and in effect there are 41 phones.

3. LETTER-TO-SOUND AND SYLLABIFICATION

Letter to sound rules for Tamil can be build using rule-based or using machine learning algorithms such as CART [8][9]. In this work we have used a set of rules to map a letter to sound and Table 1 summarizes the letter to sound rules used in our system. The consonant characters: /k/, /ch/, /r:/, /t/,



Vowels:

a	aa	i	ii	u	uu	e	ei	ai	o	oo	au	a:
அ	ஆ	இ	ஈ	உ	ஊ	எ	ஏ	ஐ	ஒ	ஓ	ஔ	ஃ

Consonants:

k	ng~	ch	nj~	t:	nd~	t	n	p	m	y	r	l	v	z	l:	r:	n:
க்	ங்	ச்	ஞ்	ட்	ண்	த்	ந்	ப்	ம்	ய்	ர்	ல்	வ்	ழ்	ள்	ற்	ன்

Fig. 1. Vowels and Consonants of Tamil along with transliteration scheme

/t:/, /p/ have different pronunciation based on the preceding and succeeding context while the remaining characters have a single pronunciation.

Table 1: Letter to Sound Rules for Tamil. [] denote nothing, * denote anything, C denote a consonant, V denote a vowel and N denote a nasal

Letter	Rule	Sound
k	[] k (*)	k
	[] k (au)	g
	(N) k (*)	g
	(a:) k (*)	g
	(y) k (*)	h
	(C) k (*)	k
	(V) k (V)	h
ch	[] ch (*)	s
	(C) ch (*)	ch
	(V) ch (V)	s
	(nj) ch (*)	j
	(N) ch (*)	s
r:	(*) r:r: (*)	tr:
	(N) r: (*)	dr:
	(*) r: (*)	r:
t	[] t (*)	t
	(*) t (C)	t
	(y) t (*)	t
	(N) t (*)	d
	(C) t (*)	t
	(V) t (V)	d
t:	(*) t: (*)	t:
	[] t: (*)	d:
	(*) t: (V)	d:
	(*) t: (C)	t:
p	(*) pp (*)	pp
	(N) p (*)	b
	[] p (*)	p
	(C) p (*)	p
	(V) p (V)	p

Typically a phone sequence in Indian languages would cluster into C^*V units, however, it is possible to have complex clusters of C^*VC^*. To handle the latter cases, we have used the following set of simple rules which were derived for after heuristic analysis of Tamil, Hindi and Telugu words.

- If there is only one consonant between the next vowel, then the consonant should go with the next vowel
- else if there are two consonants between the next vowel, then the first consonant should go with the previous vowel and the other one should go with the next vowel.
- else if there are three consonants between the next vowel, then the first two consonants should go with the previous vowel and the remaining consonants should go with the next vowel.

4. CREATION OF TAMIL SPEECH DATABASE

To build a unit selection voice, typically a small set of sentences are selected from a large text corpus such that they have good coverage of required unit such as syllable. In this work we have collected Tamil text corpus (.3 million sentences) from a news portal. This corpus has 2.7 million words and 4302 syllables. By using a greedy approach, we selected 2394 sentences which covers 25769 words and 2392 high frequency syllables.

The selected sentences were recorded by a female native Tamil speaker in a recording studio. The speaker uttered the sentences into a stand mounted microphone placed in front of her. The speech data was recorded at 44 KHz, mono channel at 16 bits per sample. After the recording it was down sampled to 16 KHz for further processing. This recording process resulted in 8 hours of speech.

5. HMM BASED SPEECH SEGMENTATION

To build a unit selection voice, the speech database has to be segmented into phones. Manual correction or segmentation is preferred but it is labor intensive and consumes time. Thus automatic segmentation tools based on machine learning techniques are often used. A comparison of dynamic programming and HMM based approach for segmentation can be found in [10]. HMMs based approach is preferred for speech segmentation and it is language independent and do not assume any knowledge such as duration of the phones. Typically Sphinx or HTK are used to perform the speech segmentation. These systems train the Hidden Markov Models using the utterances and the transcription of the unit selection voice and obtain the segmentation labels by forced-alignment of the trained data. To build the Tamil voice we used HTK to obtain the segments.

6. UNIT CLUSTERING AND SYNTHESIS

Given these segments, the unit selection algorithm in Festvox clustered the phones based on their acoustic differences. These clusters are then indexed based on higher level features such as phonetic and prosodic features. During synthesis, the appropriate clusters are sought using phonetic and prosodic features of the sentence. A search is then made to find a best path through the candidates of these clusters. Though the units used here are phones, the acoustic frames of previous unit is used during clustering as well as for concatenation. The Tamil voice built using HMM segmented labels and the unit clustering algorithm was subjected to listening test of native speakers. We synthesized 15 sentences and asked four native speakers to rank each of the utterance with a score of 1-5 (1 being very bad, and 5 being very good). The average score obtained across all utterances and speakers was found to be 3.05.

7. SEGMENTATION ACCURACY OF SPHINX AND HTK

It is well known that the quality of synthesized voice is dependent on the accuracy of the segmentation. HTK and Sphinx systems are commonly used tools for segmentation and Sphinx-II is supported in the Festvox release. We wanted to study the segmentation accuracy of Sphinx-II and HTK systems by comparing the force-aligned labels with hand corrected labels. To perform this experiment we used hand labeled speech corpus available in Hindi language. The Hindi speech database was generated for Hindi text to speech system and the labels were hand corrected by a single person trained specifically for this purpose. The Hindi database was recorded by a female speaker. The duration of this speech corpus is around 90 minutes and it has 596 utterances containing roughly 50000 phone segments.

We tried to conduct this experiment using same set of parameters in both Sphinx-II and HTK. However due to practical limitations, the setup used in this experiment is as shown in Table 2. In Sphinx-II, semi-continuous HMMs are used, skip state is allowed and context-dependent models are trained for forced-alignment. Whereas in HTK, continuous models with one Gaussian per state, left-to-right model with no skip state and context-independent models are used for forced-alignment. Both of these systems used 13 dimensional Mel-frequency cepstral coefficients (MFCC) along with delta and delta-delta coefficients. Given a segment label estimated by HMM, it was compared against hand labeled segment. The difference between estimated begin-end points and the actual begin-end points was computed and summed to call it as deviation. These values were computed for all the segments and an average deviation noted in milliseconds is used as measure of performance. The average deviation obtained by Sphinx-II and HTK are noted in the last row of Table 2.

Table 2. Parameters used and the average deviation obtained by Sphinx-II and HTK for Hindi database

	Sphinx-II	HTK
Frame Rate	10 ms	10 ms
Feature Dim.	39	39
Feature Type	MFCCs+Delta +Delta-Deltas	MFCCs+Delta +Delta-Deltas
No. of States	5	5
Skip state	Yes	No
Gaussians	Semi-Cont.	1 per state
Context-Dep.	Yes (triphone)	monophone
Avg. Deviation	**26.0 ms**	**29.59 ms**

8. SIGNIFICANCE OF DIFFERENCE PARAMETERS FOR SEGMENTATION

The difference parameters typically computed as delta and delta-delta coefficients improve the performance of speech recognition and hence are widely used [11]. We wanted to investigate the use of difference parameters for segmentation purpose. We conducted two different experiments using HTK with MFCCs, MFCCs + deltas and compared the result with that of obtained from MFCCs + delta + delta-deltas. The comparison of these three experiments is shown in Table 3. We observe that the difference parameters have no effect on the average deviation and in fact the average deviation is lesser with out the use of difference parameters. The use of difference parameters increases the dimensionality of the feature space and consume more space and time to run the experiments.

Table 3. Performance of HTK with different feature streams. D denotes delta features and DD denotes delta-delta features.

Feature Type	Avg. Deviation (ms)
MFCC	28.17
MFCC + D	28.63
MFCC + D + DD	29.59

9. RESULTS AND DISCUSSION

The results shown in Tables 2-3 indicate that the performance of Sphinx-II and HTK is similar for segmentation purposes. However, there is an average difference of 2 ms between Sphinx-II and HTK, which could be attributed to semi-continuous models or context-dependent models used in Sphinx-II. To further experiment with context-independent models we used our recently written HMM code. The use of this newly written code also served the purpose of comparing its efficiency and performance with standard tool such as HTK and Sphinx-II. The parametric setup used in this experiment is as shown in Table 4. We used two Gaussians per states and a frame rate of 5 ms. An average deviation of 24.11 ms was obtained which showed that context-independent models may be useful and relevant for speech segmentation in the context of speech synthesis as they avoid computation time and resources required to build context dependent state models.

Table 4. Performance of newly built HMM code

	HMM code
Frame Rate	5 ms
Feature Dim.	13
Feature Type	MFCCs
No. of States	3
Skip state	No
Gaussians	2 per state
Context-Dep.	monophone
Avg. Deviation	**24.11 ms**

The experimental results showed in Table 3 indicate that the delta and delta-delta coefficients may not contribute to lessen the average deviation. Typical computation of delta coefficients is done by smoothing the difference parameters and hence these coefficients could be more relevant to speech recognition that to speech segmentation.

10. CONCLUSION

In this paper, we have described the development of unit selection voice for Tamil and addressed the issue of speech segmentation problem in the context of speech synthesis. We have compared two state-of-art systems Sphinx-II and HTK with manually segmented labels and found that these systems perform similar and have produced an average deviation of around 26 ms in reference to manual labels of Hindi speech database. We also have experimented with different feature streams and observed that the difference parameters computed in the form of smoothed delta and delta-delta coefficients may not be of significant use for speech segmentation.

11. REFERENCES

[1] "Digital Library of India, http://dli.iiit.ac.in/," 2005.

[2] Raj Reddy, "PCtvt: a multifunction information appliance for illiterate people, http://www.rr.cs.cmu.edu/pctvt.ppt," in *ICT4B retreat at UC Berkeley, August 26*, 2004.

[3] S.P. Kishore and Alan W Black, "A data-driven synthesis approach for indian languages using syllable as basic unit," in *Proceedings of Eurospeech, Geneva, Switerzland*, 2003.

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

[5] Carnegie Mellon University, "SphinxTrain: building acoustic models for CMU Sphinx," http://www.speech.cs.cmu.edu/SphinxTrain/, 2001.

[6] X. Huang, F. Alleva, H.-W. Hon, K.-F. Hwang, M.-Y. Lee, and R. Rosenfeld, "The SPHINX-II speech recognition system: an overview," *Computer Speech and Language*, vol. 7(2), pp. 137–148, 1992.

[7] "Hidden Markov Model Took Kit (HTK), http://htk.eng.cam.ac.uk," 2000.

[8] N. Udhyakumar, C.S. Kumar, R. Srinivasan, and R. Swaminathan, "Decision tree learning for automatic grapheme-to-phoneme conversion for Tamil," in *SPECOM-2004, St. Petersburg, Russia*, 2004.

[9] C.S. Kumar, V. Shunmugom, N. Udhyakumar, and R. Srinivasan, "Rule-based automatic grapheme to phoneme conversion for Tamil," in *Proc. ICSLT, Delhi, India*, 2004.

[10] John Kominek, Christina Bennett, and Alan W Black, "Evaluating and correcting phoneme segmentation for unit selection synthesis," in *Proceedings of Eurospeech, Geneva, Switerzland*, 2003.

[11] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall PTR, 2001.