Using Polysyllabic units for Text to Speech Synthesis in Indian languages

Vinodh.M.V., Ashwin Bellur, Badri Narayan K., Deepali M. Thakare, Anila Susan, Suthakar N.M., Hema A. Murthy
Department of Computer Science and
Engineering
Indian Institute of Technology - Madras
Chennai 600 036
Email: {vinodh, ashwin, badri, deepali, anila, suthakar, hema} @lantana.tenet.res.in

***Abstract*—This paper describes the design and development of Indian language Text-To-Speech (TTS) synthesis systems, using polysyllabic units. Firstly, a phone based TTS is built. Later, a monosyllable cluster unit TTS is built. It is observed that the quality of the synthesized sentences can improve if polysyllable units are used (when the appropriate units are available), since the effects of co-articulation will be preserved in such a case. Hence, we built Hindi and Tamil TTS with polysyllabic units, that contains cluster units of more than one type (monosyllable, bisyllable and trisyllable). The system selects the best set of units during the unit selection process, so as to minimize the join and concatenation costs. Preliminary listening tests indicated that the polysyllable TTS has better quality.**

I. Introduction

Concatenative speech synthesis **[**1**]** systems combine sound units which are stored in a database, in order to generate the desired utterance. The advantage of using unit selection based concatenative synthesis is that there may not be a need for separate prosody modeling, because of the availability of many units under varied contexts. These sound units could be a phoneme, diphone, syllable or word etc. For building Indian language speech synthesis systems, its more appropriate to use syllables as the the basic unit. A syllable could be defined as taking the form C*VC*, where 'C' denotes a consonant and 'V' denotes a vowel. The work by Kishore and Black **[**2**]** suggests the usage of syllables as the basic unit for Indian languages. Our earlier efforts - **[**3**]**, **[**4**]**, **[**5**]**, **[**6**]**- reiterates this fact. Some of the advantages of using syllables as basic units is that they have fairly long duration when compared to phonemes or diphones. Hence, the task of segmentation becomes relatively easier. Also, since the boundaries of most of the syllables are low energy regions (due to consonants), the concatenations would result in reduced perceivable distortions.

The case for making use of poly-syllables is that, as the number of concatenation points will be less, the synthesized speech quality is expected to be better. Also, it may so happen that the Festival speech synthesis system chooses two units from different parts, even when both the units occur in sequence in the database. This would indicate that the criteria used by Festival for unit selection may be inappropriate, which could be avoided by making use of polysyllabic units. Since polysyllable units are formed using the monosyllable units already present in the database, the synthesis quality can be improved without augmenting any new set of units or realizations.

The efforts for building Indian language TTS under Festival framework includes but not limited to the work as in - **[**4**]**, **[**5**]**, **[**6**]** and **[**7**]**. The method discussed in the paper is different in the sense that the cluster units are not just built for one type of unit (monosyllable), but for different types of units (mono, bi and tri syllables). Later, we combine the results of 3 different phases (for mono, bi and tri syllables) of training appropriately to produce polysyllable TTS. Also, all of our previous efforts do not make appropriate use of pronunciation dictionary. The usage of pronunciation dictionary would greatly help in choosing the appropriate sound unit, based on context, in the case of Indian languages.

The paper is organized as mentioned below. Section II describes the design considerations involved in building a polysyllable TTS. This section includes details the steps pertaining to corpus collection, syllable coverage, building pronunciation dictionary and voice talent/artist selection process. Section III discusses the implementation aspects of building a polysyllable TTS using the Festival framework. This section also elaborates on the labeling process (using both ergodic Hidden Markov Model [EHMM] **[**8**]** and group delay **[**9**]** based segmentation algorithms) and on the multiple stages of TTS training/building, in order to produce polysyllabic speech synthesis. Section IV discusses about the possible research directions. Finally, Section V provides conclusions of this paper.

II. Design

A. Syllable Coverage

In an ideal case, the text corpus collected should contain all the possible syllables of a particular language. But, this does not generally happen since some of the syllables do not occur frequently. The paper **[**10**]** talks about the non-trivial aspect of achieving complete syllable coverage. However, the text corpus collected should provide maximum possible syllable coverage.

The implementation details of collecting text corpus for Indian languages is discussed in Section III A. Once the

text corpus is collected and processed, it is first tested for syllable coverage. The syllable set that we consider includes V, VC, CV, CVC and CCVC. The processed sentences are first segmented into syllables and their frequency and uniqueness of occurrence identified. The context in which the syllable occurs is also taken into account, to judge the uniqueness of a syllable. For example, the same syllable /hum/ in Hindi, occurring in three different contexts like begin, middle and end of a word, will be treated as three unique syllables. The sentences that contribute towards coverage of syllables are selected and are used for recording by the voice talent.

Another aspect towards improving synthesis quality is to limit the number of realizations under various contexts, thus creating a balance in the syllable corpus. This could also be achieved after the data collection process, by pruning inappropriate units.

B. Pronunciation Dictionary

Since Indian languages are syllable-centric, there needs to be a pronunciation definition for all possible contexts. The one-to-one correspondence between the written and the spoken form does not always hold good. In the case of Tamil, the same letter is used for producing different sounds under different contexts. For example, */ka/* and */ga/*, */ta/* and */da/* have the same textual representations, but their pronunciations differ with context. In the case of Hindi, in the presence of "halants", vowel deletion takes place and thus the pronunciation does not exactly match the written form. For example,

- The word सामना (*"saamanaa"*) - is textually broken down into सा (*/saa/*), म (*/ma/*), ना (*/naa/*).
- But सामना (*"saamanaa"*) - is pronounced as स् (*/s/*), आ (*/aa/*), म् (*/m/*), न् (*/n/*), आ (*/aa/*).

Also, there are cases where such vowel deletions do not take place in pronunciation. For example, the word *"prakruti"* can be split either as प्र (*/pra/*), कृ (*/kru/*), ति (*/ti/*) (or) as प्रक् (*/prak/*), ऋ (*/ru/*), ति (*/ti/*).

Thus the pronunciation dictionary for an Indian language should be capable of providing multiple pronunciations for the same word, occurring in different contexts

Our earlier efforts **[**4**] [**5**]** in building speech systems for Indian languages did not capitalize on the usage of pronunciation dictionary. The pronunciation dictionary (Lexicon) module has many advantages over the Letter-To-Sound (LTS) rules, as mentioned below :

- Pronunciation dictionary can provide appropriate pronunciations, based on context.
- Binary search is done in case of pronunciation dictionary, where as LTS uses linear search.

The pronunciation dictionary is created using unique list of words generated from the text corpus and appropriate pronunciations are provided for the same, based on their context. Another goal of using pronunciation dictionary would be to perform a Viterbi search, so that the overall concatenation cost is minimized.

C. Voice talent / Speaker selection

One hour of speech, recorded by multiple speakers was subjected to variations in pitch, speed, tempo and amplitude. This was done for both Hindi and Tamil speaker selection. The level of variations done was limited such that the quality of the voice does not change drastically. The speaker, whose voice did not loose its quality upon applying these prosodic variations, was chosen for further recording of 10 hours.

This was the major criterion for finalizing the speaker, apart from other general aspects like voice pleasantness, clarity of pronunciation and speaking rate.

D. Voice Recording

After collecting the required sentences for recording and selecting a particular voice artist, the actual recording was scheduled to be performed in stages. The collected data was recorded in a professional setting (anechoic chamber), with the following technical specifications - 16KHz sampling rate, 16 bits, single channel recording.

The recorded files, were converted to NIST sphere file format. The advantage of using sphere file format is that it can hold all the audio information as part of its header, alleviating the need for storing it separately.

III. Implementation

A. Corpus/Data Collection

The task of obtaining text corpus suitable for 10 hours of recording is not easier in the case of Indian languages. Hence, the corpus was collected by extracting text data obtained by crawling the Indian language websites from the World Wide Web. The collected data was used for recording speech waveforms for speech synthesis system building process and also to generate list of words for pronunciation dictionary.

The crawled data includes content pertaining to websites from domains like news items, story books, novels, blogs, poems, articles etc.

The Hindi corpus that was used for building TTS, initially for 1 hour, consisted of 540 sentences. 15% of these sentences were used as hold-out sentences that were used later for quality evaluation purposes (sample synthesized files can be found at **[**15**]**). These held-out sentences were chosen based on syllable coverage criterion i.e. the sentences which do not add a new unique syllable, in terms of coverage, were chosen to be part of held-out sentences. These held-out sentences hence would not affect the syllable coverage of the speech synthesis system built.

1) Labeling Tool: It is a widely accepted fact that the accuracy of labeling would have a great bearing on the quality of unit selection synthesis. The process of manual labeling is a time consuming and daunting task. It is also not trivial to label waveforms manually, at the syllable level. Our earlier work **[**4**]** required manual labeling of word boundaries, which

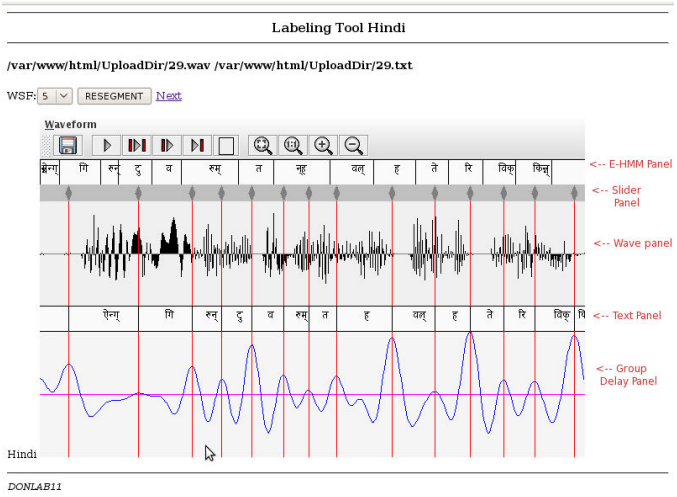

Fig. 1. Screenshot of Labeling Tool with EHMM panel

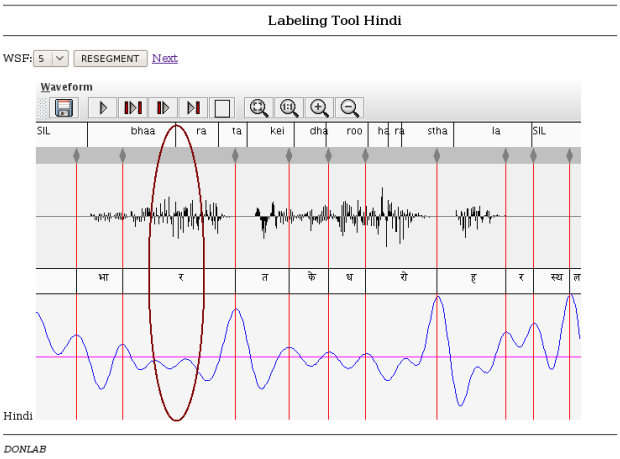

Fig. 2. Screenshot of Labeling Tool showing missed boundary

would then be used by group delay algorithm for syllable level segmentation.

DONLabel Labeling tool **[**11**]** provides an automatic way of performing labeling, given an input waveform and the corresponding text. The tool makes use of group delay based segmentation to provide the segment boundaries. The size of the segment labels generated can vary from monosyllables to poly-syllables, as the Window Scale Factor (WSF) parameter is varied from small to large values.

The previous labeling tool contains 4 panels (as can bee seen from Fig.1.) :

- Slider panel - Can be used to change/adjust the labels.
- Wave panel - Shows the waveform in segmented format
- Text panel - Shows the segmented text with syllable as the basic units
- Group Delay panel - Displays the group delay plot.

The screen shot of the labeling tool with the EHMM label information, is shown in Fig.1.

During the labeling process, we performed labeling at the entire sentence level, rather than splitting the sentences first at the word level as done in **[**4**]**. Also, our labeling process made use of both Ergodic HMM (EHMM) labeling procedure provided by Festival and the group delay based algorithm provided by the labeling tool. This was achieved by enhancing the Labeling tool to display a new panel, which would show the segment boundaries as estimated by the EHMM process. This would help greatly in adjusting the labels, if necessary, by comparing the labeling outputs of both EHMM process and group delay algorithm.

A screenshot of the labeling tool, with a missed boundary, i.e. the boundary which is not indicated by group delay, but by EHMM, is as seen in Fig.2.

A screen shot of the labeling tool, with the boundary corrected by finding the peak which lies below the threshold is shown in Fig.3. The highlighted section of Fig.3 shows the group delay peak which is missed and also the corresponding

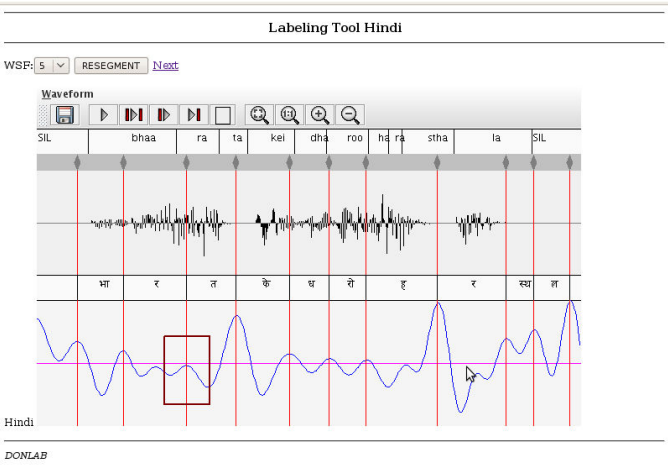

Fig. 3. Screenshot of Labeling Tool showing corrected boundary

included boundary.

As another enhancement in the labeling tool, the usage of Vowel Onset Point (VOP) detection method is being explored. By using VOP as an additional cue, manual intervention during the labeling process can be eliminated. It would also improve the accuracy of the labels generated by the labeling tool.

2) Building Poly-syllable TTS: To build a poly-syllable speech synthesis system, training was done separately thrice with different rules in the Letter-To-Sound (LTS) module. The cluster unit trees and the catalogue files were separately generated for each of the three training phases.

- Phase 1 : Monosyllable phoneset was defined and the text was broken down into only mono syllables, using LTS. The prompts and utterance structures were built for these mono syllable units.
- Phase 2 : The text was broken down into bi-syllables at the word level (where ever possible) and mono syllables if bi-syllables cannot be formed. Hence, the phoneset

included bi-syllables along with some monosyllables. Prompts and utterance structures were built for these higher level units.

- Phase 3 : The Text was first broken down into tri-syllables at the word level followed by bi and mono syllables, using appropriate LTS. The phoneset definition contained the appropriate tri, bi and mono syllable units and the corresponding prompts and utterance structures were built.

After completing the aforementioned processes, all the cluster unit trees generated by the three phases were merged into a single tree and similarly all the catalogue files were appropriately merged. This process was done to provide a large number of instances of units with varying sizes during synthesis.

The LTS rules to be specified depend on the size of the unit to be picked. For polysyllabic speech synthesis, the LTS rules are such that it first attempts to break down the text into the largest possible unit trained, at the word level. The match using the LTS rules is done linearly.

Example of LTS :
Sentence to synthesize : *maanacharitra sei aagei*
LTS Rules :

```
([ c h a r i t r a ] = charitra )
........
([ m a a n a c h a ] = maanacha )
........
([ r i t r a ] = ritra )
........
([ c h a r i ] = chari )
........
([ m a a n a ] = maana )
........
([ c h a ] = cha )
........
([ m a a ] = maa )
........
([ t r a ] = tra )
........
([ n a ] = na )
........
([ r i ] = ri )
........
```

In the above example, when synthesizing the first word *"maanacharitra"*, the LTS would pick up the first matching tri-syllable unit *"maanacha"*, followed by the bi-syllable unit *"ritra"*. Assuming that LTS entry for *"maanacha"* in not available, then the split will be as *"maana"* and *"charitra"*. Since, the LTS search is greedy in nature, it may so happen that the resulting split may not be appropriate, because of unavailability of some units.

Due to the inherent limitations of LTS, we are creating pronunciation dictionaries for Indian languages like Hindi and Tamil. The Festival framework currently allows pronunciation dictionary to be based only on word pronunciations and Parts Of Speech (POS). However, for Indian languages, we need pronunciation dictionaries that can provide pronunciations under different contexts. We need to identify, how support for the same can be incorporated in to Festival.

IV. Future Directions

A. Objective Quality Measure

Objective quality measure refers to automatic calculation of speech quality, by making comparisons on a reference signal and the signal with any form of degradation. Such measures can be helpful in reducing the cost and time involved in conducting subjective evaluation tests like MOS etc. Also, the objective measures can provide consistent results, as opposed to any subjective measure.

Some preliminary experiments on using PESQ **[**13**]** for calculating objective quality of synthetic speech, using the held-out data set, indicated that it cannot handle large time variations. Hence we plan to explore the possibility of using Dynamic Time Warping (DTW) **[**14**]** methods, which could be used for time aligning the reference and degraded signals. We can then calculate the distance between different features of the time warped signals, which could provide an indicative score of objective speech quality measure. The validity of such an objective speech quality measure would depend on the degree of correlation with the corresponding subjective Mean Opinion Score (MOS) **[**12**]**.

V. Conclusion

We have discussed the design and implementation steps involved in building a polysyllabic speech synthesis system for Indian languages using the Festival framework. The polysyllable TTS picks the largest possible unit which is available in the database. Such a criterion for unit selection synthesis helps in improving the quality, since the number of concatenation points would be greatly reduced. Also, the prosodic variations across the smaller units which make up the polysyllabic units would remain intact.

At the application level, the polysyllable TTS built was integrated with ORCA, a free and open source screen reader software for the Linux platform.

Acknowledgment

The authors would like to thank Department of Information Technology (DIT)- India, for sponsoring the project on Development of Text To Speech for Indian languages, Vide project No. CSE0809107DITXHEMA.

References

[1] T. Dutoit, *An introduction to text-to-speech synthesis*, Kulwer Academic Publishers, 1997.

[2] S.P. Kishore and A.W. Black, *Unit size in unit selection speech synthesis*, proceedings of EUROSPEECH, pp. 1317-1320, 2003.

[3] M. Nageshwara Rao,S. Thomas, T. Nagarajan and Hema A. Murthy *Text-to-speech synthesis using syllable like units*, proceedings of National Conference on Communication (NCC) 2005, pp. 227-280, IIT Kharagpur, India, Jan 2005.

[4] Samuel Thomas, M. Nageshwara Rao, Hema A. Murthy and C.S. Ramalingam, *Natural Sounding TTS based on Syllable-like Units*, proceedings of 14th European Signal Processing Conference, Florence, Italy, Sep 2006.

[5] Venugopalakrishna.Y.R. et.al., *Design and Development of a Text-To-Speech Synthesizer for Indian Languages*, pp. 259-262, proceedings of National Conference on Communication (NCC) 2008, IIT-Bombay, February 2008.

[6] Venugopalakrishna.Y.R.,Vinodh.M.V., Hema A. Murthy and C.S. Ramalingam, *Methods for Improving the Quality of Syllable based Speech Synthesis* , proceedings of Spoken Language Technology (SLT) 2008 workshop, pp. 29 -32, Goa, December 2008.

[7] Sreekanth.M and A.G.Ramakrishnan, *Festival based maiden TTS system for Tamil language*, Proceedings of 3rd Language and Technology Conference, pp. 187-191, Poznan, Poland, Oct 5-7, 2007.

[8] Alan W. Black, John Kominek, *Optimizing segment label boundaries for statistical speech synthesis* icassp, pp.3785-3788, 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009.

[9] T. Nagarajan and Hema A. Murthy, *Groupd Delay based Segmentation of Spontaneous speech into syllable-like units*, EURASIP Journal of Applied Signal Processing, Vol. 17, pp. 2614-2625, 2004.

[10] E. Veera Raghavendra, B. Yegnanarayana and Kishore Prahallad, *Speech Synthesis using approximate matching of syllables*, proceedings of Spoken Language Technology (SLT) 2008 workshop, Goa, December 2008.

[11] P. G. Deivapalan, Mukund Jha, Rakesh Guttikonda and Hema A. Murthy, *DONLabel: An Automatic Labeling Tool for Indian Languages*, pp. 263-266, National Conference on Communication (NCC) 2008, IIT-Bombay, February 2008.

[12] ITU-T Recommendations P.800, *Methods for subjective determination of transmission quality (formerly Rec. P.80*, 1996.

[13] ITU-T Recommendations P.862, *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs* , 02/2001.

[14] H. Sakoe, S. Chiba, *Dynamic programming algorithm optimization for spoken word recognition*, IEEE Transactions on Acoustics, Speech and Signal Processing In Acoustics, Speech and Signal Processing, IEEE Transactions on, Vol. 26, No. 1., pp. 43-49, 1978.

[15] Sample synthesized sentences , http://www.lantana.tenet.res.in/website_files/research/Speech/TTS/contents/main.html.