Design and Development of a Text-To-Speech Synthesizer for Indian Languages

Venugopalakrishna Y R*, Sree Hari Krishnan P†, Samuel Thomas†, Kartik Bommepally‡, Karthik Jayanthi‡, Hemant Raghavan‡, Suket Murarka‡ and Hema A Murthy*
*TeNeT Group,
Indian Institute of Technology Madras,
Chennai, India 600 036
Email: yrvenu, hema@lantana.tenet.res.in
† IDIAP Research Institute,
Martigny, Switzerland
‡Summer Interns,
IIT Madras, Chennai, India 600 036

***Abstract*—This paper describes the design and implementation of a unit selection based text-to-speech synthesizer with syllables and polysyllables as units of concatenation. The choice of syllable as a unit for Indian languages is appropriate as Indian languages are syllable-centered. Although, syllable based synthesis does not require significant prosodic modification, the prosodic modification that needs to be performed in the context of syllable is significantly different from that of conventional diphone based synthesis.**

I. Introduction

In our previous work [1] on building a voice for Tamil, it was shown that, syllables as the unit for concatenation in Festival's unit selection speech synthesizer [2] can produce natural quality speech. But Festival's implementation of unit selection synthesis does not support large or variable sized units completely [3], as it is primarily designed for smaller units like diphones, phones and half phones. This was a bottle neck for research in syllable based unit selection synthesis. Therefore, there was a need for a TTS engine which is primarily designed to handle syllable units. In this paper, we discuss the design and implementation of a new TTS based on unit selection synthesis approach, with syllables as units. The system is designed such that, it is robust enough to be a real world synthesizer and flexible enough to be a research tool.

In unit selection speech synthesis, a speech database is designed such that each unit is available in various prosodic and phonetic contexts. The speech database is considered as a state transition network with each unit in the database occupying a separate state. The state occupancy cost (target cost) is the distance between a database unit and a target unit, and the transition cost (join cost) is an estimate of the quality of concatenation of two consecutive units. A pruned Viterbi search is used to select the best unit sequence, which has lowest overall cost (weighted sum of target cost and join cost) [4].

The system as in Fig. 1 comprises of text processing, unit selection, prosody prediction, and concatenation modules. Text processing module breaks incoming text sentence to a syllable sequence. Unit selection module selects the best unit realisation sequence from many possible unit realisation sequences for the given syllable sequence. Prosody prediction module predicts energy, pitch etc.. Finally, in concatenation module the units are modified according to the predicted prosody and are concatenated. Sections II, III, IV, and V give complete description of design of these modules. Section VI discusses the design and implementation of database.

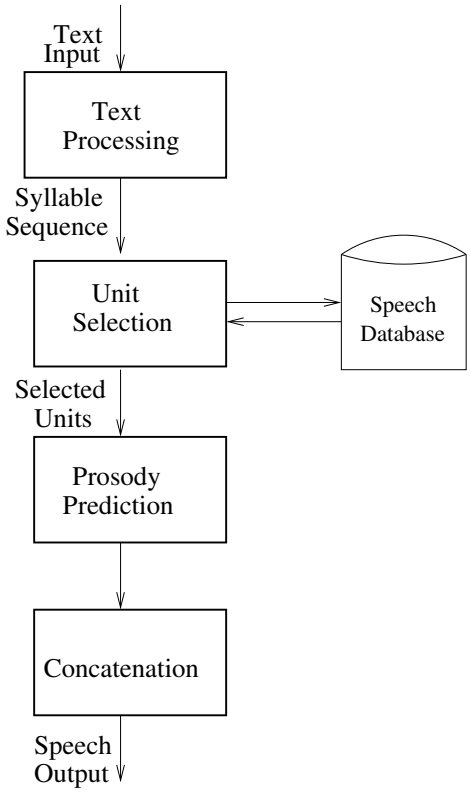

Fig. 1. Block diagram of TTS

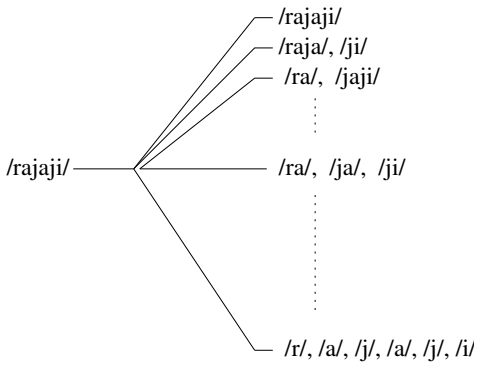

Fig. 2. Syllabification Procedure

II. Text Processing

Text input to the synthesizer can be in transliterated form or in UTF-8 [5] form. If incoming text is in UTF-8 form it will be converted to the transliterated form before further processing. The Text processing module consists of preprocessing and syllabification modules. The text in transliterated form is preprocessed to remove invalid characters in the text. And also, preprocessing module adds phrase break indicators to the text based on full stops and case markers. The preprocessed text is further passed on to the syllabification module.

A. Syllabification

It is hard to cover all the syllable units of a language in the database. And also it is not possible to cover all of them in various contexts. So, there will be a need to handle the case of missing units. Considering this, two approaches of syllabification are used.

In the first approach, the syllabification algorithm breaks a word such that there are minimum number of breaks in the word, as minimum number of joins will have less artefacts. The algorithm dynamically looks for polysyllable units making up the word, cross checks the database for availability of units, and then breaks the word accordingly. If polysyllable units are not available, then algorithm naturally picks up smaller units. This mean, if database is populated with all available phones of language alongwith syllable units, algorithm falls back on phones if bigger units are not available. For example, as in Fig. 2, for breaking a word "rajaji" algorithm looks for unit "/rajaji/" in database, if not found it looks for unit combinations such as "/raja/, /ji/", "/ra/, /jaji/" etc.. This way, it finally falls back on phone sequence "/r/, /a/, /j/, /a/, /j/, /i/".

In the second approach, the syllabification algorithm [6] breaks a word into monosyllables without checking for its availability in the database. Here syllabification is done based on standard linguistic rules. By this method "rajaji" is broken as "/ra/, /ja/, /ji/". If a unit is not found it can be substituted by a nearest unit or by silence.

III. Unit Selection

Unit selection module is responsible for selecting the best unit realisation sequence from many possible unit realisation sequences from the database. Basic cost measures, target cost and join cost [7] were used in searching for the best unit sequence. On using syllables as units, phoneme centric target features like phoneme type, place of articulation etc. used in Festival loose their meaning. Features such as position of the syllable in the word (begin, middle and end), position of the syllable in the sentence are important and can be used in target cost evaluation. As syllables are prosodically rich units, using them in appropriate position of the word is very important. In this implementation, instead of using position of the syllable in the word in target cost, we have pre-classified units according to position of the syllable in the word as begin<syl>, mid<syl> and end<syl>. During unit selection, the units are picked based on this classification. Begin<syl> correspond to unit <syl> obtained from the beginning of a word, mid<syl> correspond to unit <syl> obtained from the middle of a word and end<syl> correspond to unit <syl> obtained from the end of a word.

Join cost is the measure of how good is the joining between two consecutive units. MFCC based spectral distance (euclidean) (1) measure between last frame of one unit and first frame of the next unit is used in evaluating the join cost.

$$C_s^c(u_{i-1}, u_i) = \sum_{j=1}^{N} \left(X(j) - Y(j)\right)^2 \tag{1}$$

where N is dimension of MFCC vector, $\bar{X}$ is MFCC vector of the last frame of $(i-1)^{th}$ unit u_{i-1} and $\bar{Y}$ is MFCC vector of the first frame of i^{th} unit u_i.

Other prosodic features like pitch and energy can also be used in evaluating join cost. In such cases, the join cost, given weights w_j^c, is calculated as follows:

$$C^c(u_{i-1}, u_i) = \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) \tag{2}$$

where q is number of prosodic features and C_j^c is join cost of j^{th} prosodic feature.

Viterbi search algorithm is used to find the unit sequence with minimum overall cost. Unit selection procedure for synthesizing a word "rajaji" from various realizations of units /ra/, /ja/ and /ji/ is shown in Fig. 3.

IV. Prosody Prediction and Modification

In prosody prediction module prosodic features like energy, pitch etc. are predicted for the selected syllables. During recording of prompts, the prosody with which voice talent reads the prompt varies over the length of the recording. In addition, syllables used in concatenation are picked from different contexts. Because of these reasons, audible discontinuity due to discontinuous prosodic contours is perceived in the synthesized speech. To correct these prosodic contours, Classification and Regression Tree (CART) [8] is used in predicting prosody for the selected units.

The construction of CARTs has become a common basic method for building statistical models from simple feature

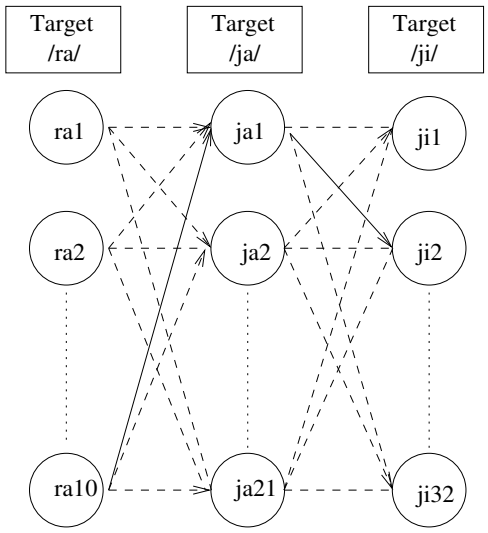

Fig. 3. Unit Selection Procedure showing selection of units /ra/, /ja/, /ji/

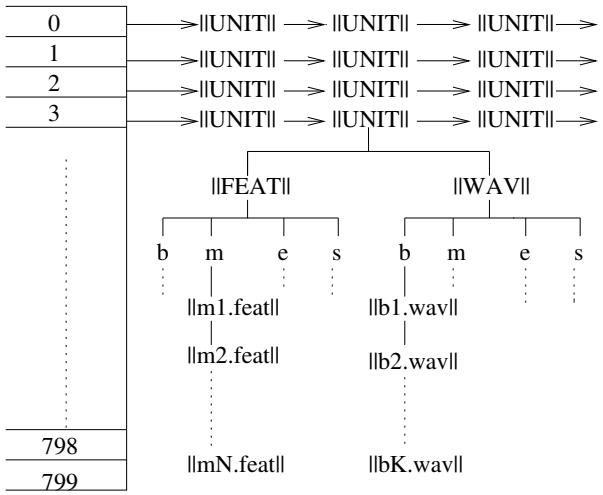

Fig. 4. Pictorial view of Database

data. CART is powerful because it can deal with incomplete data, multiple types of features (floats, unumerated sets) both in input features and predicted features, and the trees it produces often contain rules which are humanly readable. Wagon [9], a tool part of EST (Edinburgh Speech Tools) library is used to build CART.

For example, CART was built for predicting energy for the units. We used syllable name, previous syllable name, next syllable name, position of the syllable in the word (begin, end, middle, single) and normalized position of the word containing syllable in the sentence (float value from 0 to 1) as the input features for predicting peak amplitude level of the syllable unit. CART was built from a data set made up of 450 sentences corresponding to 45 minutes of speech. The correlation coefficient which is indicative of how well CART predicts the expected energy was 0.91. CART is used to predict the peak amplitude of each of the syllables in the sentences. The syllable waveforms are then scaled appropriately in concatenation module.

V. Waveform Concatenation

Selected speech units are modified according to the predicted prosody and concatenated to form a single speech file. TD-PSOLA [10] algorithm is implemented to scale pitch and duration. Energy of the syllable is also scaled based on predicted value of peak amplitude.

Apart from waveform concatenation, linear prediction based speech synthesis module is also provided as an optional module. LP coefficients and residual are precomputed for speech units in the database and later used in producing the synthetic speech for selected units. When spectra at unit boundaries need to be modified, LP synthesis can be a useful technique. This is yet to be implemented in the synthesizer.

VI. Database Design

The sentences were designed from DBIL [13] and consists of 1180 prompts. Later, prompts were recorded in a near anechoic chamber by a voice talent. Recorded prompts were manually labeled at word level and then they were segmented and labeled into syllable-like units using Group-Delay based segmentation algorithm [11], [12]. Syllable segments from continuous speech database are extracted and classified into begin, mid, end and single units based on their position in the word. Each syllable segment has a corresponding feature file describing unit's phonetic and prosodic context. Details in this feature file can be used in target cost evaluation.

In this implementation, TTS on initialisation, loads entire database containing syllable segments and their feature descriptions into a data structure. The data is stored in a hash table. Every syllable unit is hashed into one of the 800 buckets of the hash table. Syllable segments and their feature descriptions are stored using linked lists under hash list. A pictorial view of data structure used is shown in Fig. 4.

VII. Conclusion

We have discussed design and development of a text-to-speech synthesizer for Indian languages. Our design is centered around using larger or variable sized units (syllables) in synthesis. We have pre-classified units according to their position in the word. This improves synthesis quality and it reduces search space improving the synthesis timing. Database design based on hash tables also reduces search time. We need to use other phonetic features like identity and position of previous, next units in target cost evaluation. Alongwith this, prosodic descriptions such as average pitch, duration and energy must also be used in target cost. Although LP based synthesis is implemented, it is not clear at the time of writing, how spectral interpolation will be performed at the boundaries. Web interface for Hindi voice on our TTS is available at URL http://lantana.tenet.res.in/apache2-default/Research/Speech/TTS/DonlabTTS/Transliteration1.php.

ACKNOWLEDGMENT

The authors would like to thank IBM for funding the summer interns vide project no: CSE/06-07/087/IBMC/HEMA.

REFERENCES

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

[2] The Center for Speech Technology Research, The University of Edinburgh, http://www.cstr.ed.ac.uk/projects/festival/.

[3] Robert A.J. Clark, Korin Richmond, and Simon King, "Festival 2 - build your own general purpose unit selection speech synthesiser", In Proc. 5th ISCA workshop on speech synthesis, 2004.

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

[5] F. Yergeau, "UTF-8, a transformation format of ISO 10646", November 2003.

[6] Lakshmi A. and Hema A. Murthy, "A Syllable based continuous speech recognizer for Tamil", Interspeech 2006, ICSLP, Pittsburgh, Sep. 2006.

[7] A.W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis", Proceedings of EUROSPEECH, 1997, pp. 601–604.

[8] L.J. Breiman, H.Friedman, R.A. Olshen and C.J. Stone, "Classification and Regression Trees".

[9] A.W. Black and K.A. Lenzo, "Building synthetic voices", 2003, http://festvox.org/bsv/

[10] E. Moulines and Charpentier F., "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones", specom, 1990, pp. 9(5/6):453-467.

[11] T. Nagarajan, H. A. Murthy and R. M. Hegde, "Segmentation of speech in to syllable-like units", in proceedings of EUROSPEECH, 2003, pp. 2893-2896.

[12] T. Nagarajan and H. A. Murthy, "Group delay based segmentation of spontaneous speech into syllable-like units", EURASIP Journal of Applied signal processing, vol. 17, pp.2614-2625, 2004.

[13] "Database for Indian Languages", Speech and Vision Lab, IIT Madras, India, 2001.