IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 4, Ver. II (Jul-Aug. 2014), PP 49-57
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

An Unit Selection based Hindi Text To Speech Synthesis System Using Syllable as a Basic Unit

Shreekanth.T [1] Udayashankara.V [2] Arun Kumar.C [3]
[1] *(Research Scholar, JSS Research Foundation, Mysore, India)*
[2] *(Professor, Department of IT, SJCE, Mysore, India)*
[3] *(PG Scholar, Department of ECE, SJCE, Mysore, India)*

Abstract: *Concatenative speech synthesis using phoneme, di-phone and allophone as an elementary unit for Hindi speech synthesis requires significant quality improvement. The naturalness of the state of the art waveform synthesizer is attributed due to the use of syllable as a basic unit. The primary reason for choosing the syllable as a basic unit is that the Indian languages are syllable centered. This work proposes a syllable based speech unit for concatenative speech synthesis considering position of syllable in a word into account i.e the start, middle and end. This is achieved by building a standard syllable (C*V) level speech database consisting of 442 syllables in each position thus accounting for 1326 standard and non-standard words. Further the quality of synthesized speech is enhanced using moving average windowing. The effectiveness of the system is demonstrated by synthesizing natural sounding speech for Hindi, national language of India. An important advantage of this approach leads to reduced prosody mismatch and spectral discontinuity that occurs during syllable concatenation. The results obtained from the proposed system are far superior compared to the traditional unit based Text to Speech (TTS) synthesis system. The most important quality of this system is the improved naturalness in the synthesized speech.*

Keywords: *Concatenative synthesis, Hindi, Mean Opinion Score (MOS), Praat, TTS, UNICODE.*

I. Introduction

Text to speech synthesis is a technology to convert an arbitrary input text to intelligible and natural sounding speech in an attempt to transmit information from a machine to a person. The conversion of words in written form into speech is nontrivial. Moreover, in order to sound natural, the intonation of the sentences must be appropriately generated. Synthesis of speech cannot be accomplished by cutting and pasting smaller units together. Attention has to be paid to smoothing out the discontinuities in such a process so that the resulting signal approximates natural speech. According to the speech generation model used, speech synthesis can be classified into three categories as Articulatory synthesis, Formant synthesis and Concatenative synthesis [5].

Articulatory synthesis, attempts to model the human speech production system directly.Formant synthesis, models the pole frequencies of speech signal or transfer function of vocal tract based on source-filter model and Concatenative synthesis, uses unalike lengths of prerecorded samples derived from natural speech [6].

Concatenative synthesis simply plays back the waveform with the matching phone string. An utterance is synthesized by concatenating together several speech fragments, unlike synthesis-by rule; it entails neither rules nor manual tuning. Furthermore, each segment is completely natural, so we ought to expect a very natural output. Speech segments are greatly affected by co articulation, consequently if we concatenate two speech segments that were not adjacent to each other, there can be spectral or prosodic discontinuities. Spectral discontinuities occur when the formants at the concatenation point do not match while on the other hand Prosodic discontinuities occur when the pitch at the concatenation point do not match. A listener rates a synthetic speech as poor when it contains large discontinuities, in spite of each segment being very natural. There are a number of factors which contribute to the lack of naturalness in the speech output from speech synthesis systems like: Intonation and rhythm, variability along the prosodic parameters and incorrect segmental rendering. Prosody and intonation are the factors affecting the natural sounding speech [5].

Text to speech synthesis system developed for Tamil using group delay based segmentation algorithm results in high quality speech with mean opinion score of 4.1 [1].Concatenative based Text To Speech for Hindi, Tamil and Bengali language using di-phone as basic unit reduces junction mismatch and time delay during concatenation [2].Symbol based concatenation approach for TTS system for Hindi language using vowel classification technique results in natural speech output with mean opinion score of 4.3 [3].Unit selection based TTS system developed for Indian language will provide the clear idea of different choices of units for concatenative synthesis [4].

Concatenation based text to speech system developed using above algorithms also results in large concatenation points. This large concatenation results in glitch at the output which is hard to eliminate prosody mismatch and spectral discontinuity. These spectral discontinuity and prosody mismatch reduces the naturalness of concatenated speech output. This encouraged us to build a syllable level text to speech system. Hence before entering into concatenative speech synthesis, speech database building plays a vital role in achieving the goal of attaining natural sounding speech. With an aim of achieving this, authors have developed a high quality syllable level speech database for Hindi language taking the position of syllable into consideration [8].

After developing the speech database another significant decision to make is to select a more accurate algorithm amongst the concatenative speech synthesizer. From various researches and experiments direct waveform concatenation technique is more suitable among all the concatenative speech synthesis [7].

This algorithm gives very high quality speech output when compared to other synthesizing techniques, as it uses the syllable units considering the position of the syllable in a word into account. This paper is divided into four sections: In section II an overview of Hindi language and its properties are presented. Section III discusses in detail about the development of proposed speech database. Section IV presents the results and discussions. Section V, provides conclusions and directions for the future work.

II. Overview Of Hindi Language

Hindi is the official language of India, it is spoken as a first language by 33 percent of the Indian population, and by many more as a lingua franca. In contrast only a very small percentage of Indians use English as a means of communication. This fact coupled with the prevalent low literacy rates makes the usage of conventional user interface challenging in India.

In Hindi language there are 33 consonants and 13 vowels. Hindi language is having one to one correspondence with the spoken language and the written form. The phonemes are divided into two type vowels (swaras) and consonants (vyanjanas) [8].

Vowels (Swaras): Vowels are the independently existing letters which are called swaras. Vowel sound cannot be modified.

अआइईउऊऋएऐओऔ

Consonants (Vyanjanas): Consonants are those which depend on vowels to form their independent letter. Consonants sound can be modified by combining vowels with consonants.

कखगघङ, चछजझञ, टठडढण, तथदधन, पफबभम, यरलवशषसह

Therefore we can say that Hindi language is phonemic in nature. Amalgamation of vowels with consonants will from a syllable and it is also called as "Kagunitha".

III. Speech Database Building

During the process of speech synthesis the required speech units are fetched from database, concatenated and further processed using a suitable algorithm. Hence creating an error free database considering syllable as a basic unit is of greatest important.

In order to carry out this task, a set of about 1540 words were collected from standard 'Hindi to English dictionary' [11]. Later speech recording is done using utility software for windows operating system, called as the Praat [9]. The syllables are recorded with a sampling frequency of 16 KHz and represented using 16-bits. The pitch and formant frequency of the syllable fluctuates with position of the syllable in uttered speech. Thus the syllable level database is developed for all the possible position of occurrences. The syllable can befall at three possible positions.

1. Beginning of the word (Start)
2. Between the two syllables (Middle)
3. At the end of the word (End)

To get natural sounding at the synthesized speech, each syllable is extracted from a word containing syllable. Three words were selected such that, it should contain the required syllable in all the possible positions. Then the necessary syllables are extracted from that particular position using Praat tool and stored in directories named start, middle and end as shown in Fig.1 with a unique numerical value based on their Unicode [8].

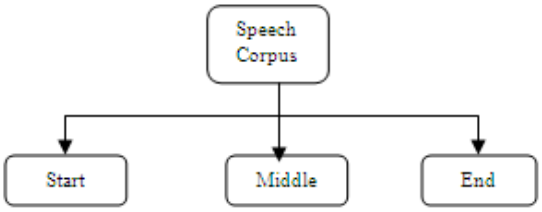

Fig.1. Sound library start, middle and end

IV. Methodology

Text to speech is to transfigure an orthographic text to its acoustic signal. The main process involved in building the TTS systems are, Text processing, Speech corpus building, and Speech synthesis. Fig.2 spectacles the block diagram of TTS system. This section covers in brief about the steps involved in building TTS system and rules applied during pre- processing of input text.

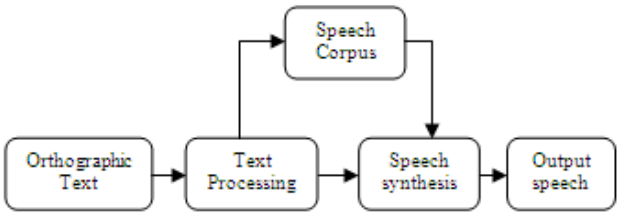

Fig.2. Block diagram of Text to Speech synthesis

Once the orthographic text is entered as input to text to speech system, text processing unit acts as front end of TTS system. Speech corpus unit consists of syllable level speech database which is segmented and labeled based on text processing output. Concatenative speech synthesizer comprises of a search algorithm, silence removal algorithm and a speech quality improvement algorithm.

Text processing is done inorder to convert the orthographic text to its corresponding Unicode and it is stored in a text file for further analysis. The Unicode output is used as an input to concatenative speech synthesizer. The search algorithm fetches the Unicode output one after the other and matches it each time. After that the wave file is fetched from speech corpus and it is concatenated. Finally the concatenated speech output is passed through a silence removal algorithm to eliminate the external noise present in the concatenated speech.

A. Text Processing

Text processing is the first and foremost step involved in developing TTS system. The main intension of this stage is to convert the input text into its equivalent Unicode and later it is mapped to a unique decimal code. The Hindi symbols and alphabets are congregated into diverse classes as revealed in TABLE I to VI. Based on the above assumptions an algorithm is developed using .NET programming language.

Table 1. Independent VOWEL

Alphabets	Unicode	Decimal Equivalent
अ	0905	2309
आ	0906	2310
इ	0907	2311
ई	0908	2312
उ	0909	2313
ऊ	090A	2314
ऋ	090B	2315
ए	090F	2319
ऐ	0910	2320
ओ	0913	2323
औ	0914	2324

Table II. Consonants

Alphabets	Unicode	Decimal Equivalent
क	0915	2325
ख	0916	2326
ग	0917	2327
घ	0918	2328
ङ	0919	2329
च	091A	2330
छ	091B	2331
ज	091C	2332
झ	091D	2333
ञ	091E	2334
ट	091F	2335
ठ	0920	2336
ड	0921	2337
ढ	0922	2338
ण	0923	2339
त	0924	2340
थ	0925	2341
द	0926	2342
ध	0927	2343
न	0928	2344
प	092A	2346
फ	092B	2347
ब	092C	2348
भ	092D	2349
म	092E	2350
य	092F	2351
र	0930	2352
ल	0932	2354
व	0935	2357
श	0936	2358
ष	0937	2359
स	0938	2360
ह	0939	2361

Table III. Dependent Vowel Sign

Alphabets	Unicode	Decimal Equivalent
◌ा	093E	2366
◌ा	093F	2367
◌ि	0940	2368
◌ी	0941	2369
◌ु	0942	2370
◌ू	0943	2371
◌े	0947	2375
◌ै	0948	2376
◌ो	094B	2379
◌ौ	094C	2380
◌ॅ	094D	2381

Table IV. Padding

Alphabets	Unicode	Decimal Equivalent
◌	093E	---
◌ा	093F	01
◌ि	0940	02
◌ी	0941	03
◌ु	0942	04
◌ू	0943	05
◌े	0947	06
◌ै	0948	07
◌ो	094B	08
◌ौ	094C	09
◌्	094D	10

Table V. Numbers

Numbers	Unicode	Decimal Equivalent
0	0966	2406
1	0967	2407
2	0968	2408
3	0969	2409
4	0970	2410
5	0971	2411
6	0972	2412
7	0973	2413
8	0974	2414
9	0975	2415

Table VI. Special Symbols

Symbols	Unicode	Decimal Equivalent
◌ँ	0901	2305
◌ं	0902	2306
◌ः	0903	2307
ॐ	0950	2384
॰	0950	2416
।	0964	2404
॥	0965	2405

The program reads the entered text, character by character and generates the modified Unicode as the output. The algorithm for this is described below.

Algorithm-1:

1. The algorithm will check for, whether the user has entered the input text or not by computing the length of the entered text.
2. If the entered text length is zero then algorithm decides that no text has been entered, else if the text length is non-zero then the program first calculates the total length of the entered text.

3. Later, the program splits the entered text into sentences and sentences into words by spotting spaces present between every word in a sentence.
4. The segmented words are then stockpiled in a memory unit for further analysis. The stowed words are chosen one after the other and are mapped with an array element which encompasses "English transliteration codes" of Hindi language.
5. After the process of transliteration the output file is stored in a separate text file.
6. The warehoused result is read character by character to check whether it belongs to the vowel (V) group or the consonant (C) group.
7. If the read character fits in to the vowel group then its Unicode is directly mapped from the Table I. E.g. Consider character read is अ its decimal code is 2309 and this is padded with two zeroes directly. The modified Unicode will be 230900.
8. If the read character belongs to the consonant group as shown in Table II then algorithm checks for the following character and if that character also belongs to consonant group then map the previous consonant directly from Table II. E.g. consider the character read is ण its decimal code is 2339 and the next character read is also a consonant then Unicode remains the same 2339.
9. If the read character belongs to the consonant group as shown in Table II then algorithm checks for the following character and if that character belongs to the dependent vowel sign as in Table III then Unicode of that consonant is padded with corresponding two digit values as shown in Table IV. E.g. Consider character entered is रु then it is divided into र its decimal code is 2352 and ु its code is 2370, its padding value obtained from Table IX is 04. So modified Unicode value is 235204.
10. If the read character belongs to a special character or a number (N) then the Unicode is directly mapped from Table V and VI.
11. The entire process continues until the end of the text is reached. The modified Unicode is used for further processing.

The output of the proposed algorithm for the entered text अरुण andअरुणकुमारis described below:

If the entered word is अरुण then its Modified Unicode output will be 230900 235204 2339, the presence of spaces between each Unicode helps us to differentiate the individual characters in the entered word.

If the entered sentence is अरुणकुमार then its Modified Unicode output will be 230900 235204 2339 101010 232504 235001 2352. Unicode 101010 acts as a space between two words and is used to differentiate two words during sentence formation.

B. Concatenative Speech Synthesis

The concatenation algorithm developed here is the furtherance of the text processing phase presented in the preceding section. It consists of a search algorithm which assents the text file generated by text processing algorithm as input and gives concatenated speech as the output. Based on the text processing result developed algorithm detects the corresponding audio file present in the speech database and concatenates the individual audio files. The algorithm is as described below.

Algorithm-2:

This algorithm has been implemented using the MATLAB tool. The text file generated from text processing phase is imported to the MATLAB directory. As soon as the text file is available on MATLAB directory the concatenation code reads the input text processed file 'X' and calculates the length of that text file 'L'. After calculating the length of text file, the algorithm first checks weather the user has entered a word or a sentence by detecting the number of spaces present in the entered text. The space is identified by a decimal value 101010. If the space is present, in the entered text, then it will consider it as a sentence and then stores the position of 101010 detected into a text file 'M'.

Later the text file 'M' is read to detect the length of each word present in a sentence. Steps involved in detecting the word length are shown below.

1. If the detected word length is 1 i.e. C=1, then the algorithm compares the text file output a=X (i) with the front speech database and extracts corresponding audio file and concatenates.
2. If word length is 2 i.e. C=2, then the algorithm compares a=X (i) and a1=X (i+1) with the front and back databases respectively, and concatenates.
3. If word length is 3 i.e. C=3, then the algorithm compares a=X (i), a1= X (i+1) and a2=X (i+2) with front, middle and back databases respectively, and concatenates.

4. If word length is greater than 3 then algorithm displays a message 'Limit exceeded'.

The above mentioned steps is repeated for the whole text. Finally the concatenated speech signal is passed through a silence removal algorithm to eliminate the delay present between each concatenation point.

C. Silence Removal

Synthesized speech signals usually contains areas of silence or noise. It is necessary to remove the said components, in order to detect "clean" speech segments. The silence removal algorithm [10] implemented in this work is based on two basic audio features, namely the signal energy and the spectral centroid.

1. Signal Energy: Let x_i (n); n = 1, 2… N be the audio samples of the i^{th}frame. Then, for each frame i, the signal energy $E_i$is given by,

$$E(i) = \qquad (1)$$

2. Spectral Centroid**:** The spectral centroid, C_i of the i^{th} frame is defined as the center of gravity of its spectrum.

$$C_i = \qquad (2)$$

where, X_i (k), k = 1, 2... N, are the Discrete Fourier Transform (DFT) coefficients of the i^{th} frame. N is the frame length. Ci is a measure of the spectral position. High values of C_i correspond to "clear" sounds.

a. Signal energy and spectral centroid are extracted for the entire audio signal by isolating the signal into non-overlapping frames.
b. For each sequence two thresholds are dynamically estimated.
c. Speech segments are detected based on the above criterion.

This algorithm will remove the unwanted speech units present in the concatenated speech fragments. Lastly the concatenated speech is passed through a moving average hamming window to smoothen the speech fragments.

V. Results And Discussion

The results of this work is evaluated to check the naturalness of the synthesized speech on the two sets ofdeveloped databases, one considering syllable position (DB1) andthe other without consideringthe position of the syllable (DB2). To verify the naturalness of the synthesized speech obtained from these two databases Mean Opinion Score (MOS) is taken from various listeners.

The output of the text processing phase is shown in a Graphical User Interface (GUI) as shown in Fig 3. This will give the user a vibrant idea of text processing and its output.

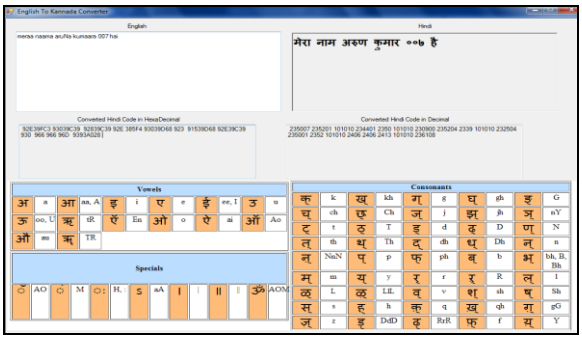

Fig.3. Text processing GUI

The GUI shown in Fig 3 consists of four sub-windows, where the first window is user controlled, second one displays entered transliterated Hindi text and the third one displays the Unicode output while the final window displays the decimal output. Further processing is done using MATLAB for speech synthesis. The output of concatenative speech synthesis, short time energy calculation and silence removal algorithm are shown in Fig 4, 5 and 6 respectively. The final silence removed output is passed through a moving average hamming window and the output is as shown in Fig 7.

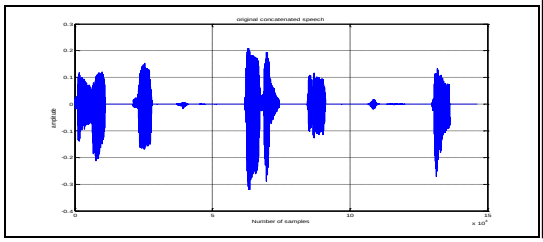

Fig.4. Concatenated speech units

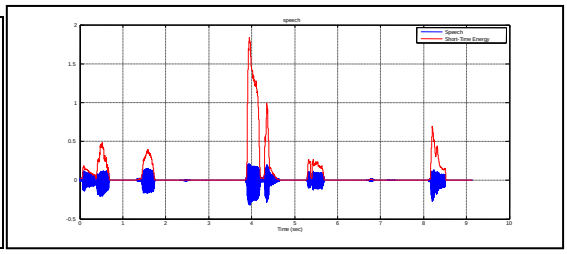

Fig.5. Short time energy calculation

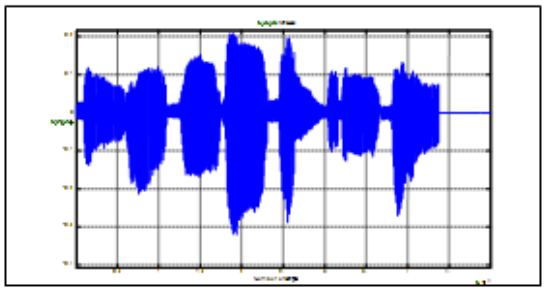

Fig.6. Silence removed speech units

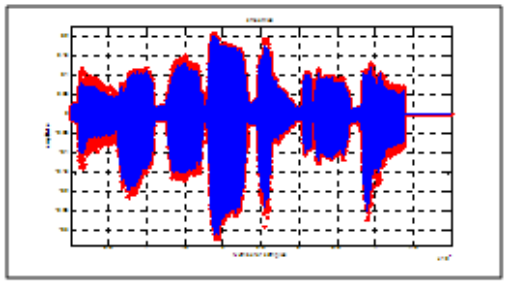

Fig.7. Moving average hamming window output

Mean opinion score is the arithmetic mean of all the individual scores and it gives the numerical indication of the perceived audio quality. To check the naturalness of the concatenated speech a listeners test was conducted to evaluate the results of the proposed technique. The algorithm is tested on both the databases DB1 and DB2 in order to compare the results. Seven listeners were asked to rate the quality of the speech synthesized using the proposed technique for both the databases. Each listener were asked to rate on a scale from 1 to 5, where 1 represents the lowest perceived audio quality where as 5 represents the highest perceived audio quality, as shown in Table VII.

Table 6. Parameter for Mean Opinion Score

MOS	Quality	Distortion
5	Excellent	Imperceptible
4	Good	Slightly imperceptible
3	Fair	Slightly Annoying
2	Poor	Annoying
1	Bad	Very Annoying

After all the initial assumptions to check the quality of the synthesized speech is done, list of words are entered in the text editor and the program is checked for both DB1 and DB2. MOS is taken from all the listeners for the synthesized speech from both the databases for the individual words and is as shown in Table VIII.

Table 7. Mos Obtained From Listeners

SR.No	Tested	L1		L2		L3		L4		L5		L6		L7		Total	MOS	MOS
	Words	DB1	DB2	DB1	DB2	DB1	DB2	DB1	DB2	DB1	DB2	DB1	DB2	DB1	DB2	Listeners	DB1	DB2
1	कमली	4.5	4.0	4.5	3.5	4.0	3.5	4.0	3.5	4.5	4.0	4.5	4.0	4.5	4.0	7	4.36	3.78
2	कोमल	4.5	3.5	4.5	4.0	4.5	3.5	4.5	3.0	5.0	4.0	4.5	4.0	4.5	4.0	7	4.57	3.71
3	मेरी	4.5	4.0	5.0	4.0	4.0	3.0	4.0	3.5	4.5	3.5	4.0	3.5	5.0	4.0	7	4.43	3.64
4	हमारा	3.5	3.0	4.5	3.0	4.5	3.5	4.5	3.0	4.5	4.0	4.5	3.0	4.5	4.0	7	4.36	3.35
5	मामी	4.5	4.0	5.0	4.0	5.0	4.5	4.5	4.0	4.5	3.0	5.0	4.0	5.0	4.0	7	4.78	3.93
6	पारु	5.0	3.5	5.0	4.0	4.5	4.0	4.0	3.5	4.0	3.5	4.5	3.5	4.5	3.5	7	4.51	3.64
7	पानी	4.5	4.0	5.0	4.0	4.0	4.0	2.0	4.0	4.5	4.0	4.5	4.0	5.0	4.5	7	4.21	4.07
8	साडी	4.5	4.5	5.0	4.0	5.0	4.0	4.5	4.0	4.5	4.0	5.0	4.0	4.5	4.0	7	4.57	4.07
9	नवीना	5.0	5.0	5.0	4.5	4.0	4.0	4.0	3.0	5.0	4.5	4.5	4.0	4.5	3.5	7	4.57	4.07
10	पावना	4.5	4.0	4.5	4.0	5.0	4.0	4.5	3.5	4.5	4.0	4.5	4.0	4.5	4.0	7	4.57	3.93
Average MOS																	4.493	3.819

VI. Conclusion

Text to Speech synthesis system developed using the unit selection based concatenative speech synthesis is productively implemented using .NET and MATLAB programming languages. Spectral discontinuity and prosody mismatch is minimized by means of standard syllable level speech database. The quality of the synthesized speech is evaluated by various listeners by taking the MOS. From the evaluation results the MOS obtained for DB1 is 4.49 and for DB2 it is 3.85. Hence from these results we can conclude that the TTS system developed using database DB2with syllable as a basic unitconsidering the position of syllable in a word, provides improved results when compared to other traditional units. The appealing characteristic of the proposed TTS system is, with little modifications this work can be extended for most of the Indian languages.

Acknowledgement

We would like to thank Council of Scientific and Industrial Research (CSIR), New Delhi, India for providing the financial backing for this work under the research scheme No. **22(0613)/12/EMR-II** and also the work is supported by JSS Research Foundation, Mysore, Karnataka, India.

References

Journal papers:

[1]. Thomas, Samuel, M. Nageshwara Rao, Hema A. Murthy, and C. S. Ramalingam. "Natural sounding TTS based on syllable-like units." Energy 2, no. 4, 2006.

[2]. Sangamitra Mohanty, "Syllable Based Indian Language Text To Speech System",IJAET, Vol. I, Issue 2, 2011, pp. 138-143.

Proceedings papers:

[3]. Pamela Chaudhury, Madhuri Rao and K Vinod Kumar, "Symbol Based Concatenation approach for Text to Speech System for Hindi using Vowel classification technique", NaBIC 2009, pp. 1082-1087.

[4]. S P Kishore & Alan W Black, "Unit Size in Unit Selection Speech Synthesis", Indian Institute of Information Technology, Hyderabad & ISRI Carnegie Mellon University. Eurospeech 2003.

Journal papers:

[5]. Ravi D J and Sudarshan Patilkulkarni, "A Novel Approach to Develop Speech Database for Kannada Text-to Speech System", Int. J. on Recent Trends in Engineering & Technology, Vol. 05, No. 01, 2011.

Theses:

[6]. Lemmety S, "Review of Speech Synthesis Technology", M.S. Thesis, Department of Electronics and Communication Engineering, Helsinki University of Technology, 2009.

Journal papers:

[7]. Arun Kumar C and Shreekanth T , "A Comprehensive review on Concatenation Based Text to Speech Synthesis for Indian Language",IJEETC, Vol. 3, No. 2, ISSN 2319 – 2518, 2014.

[8]. Arun Kumar C, Shreekanth T and Udayashankara V, "Development of Speech Database for Hindi Text to Speech system Considering Syllable as a Basic Unit" International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5. ISSN: 2277 128X, 2014.

Software Tool:

[9]. Boersma and Weenik," PRAAT: A tool for phonetic analysis and sound manipulations"1992-2001, www.praat.org.

Theses:

[10]. T. Giannakopoulos, "A method for silence removal and segmentation of speech Signals, implemented in Matlab", Ph.D. thesis, University of Athens, Greece, 2009.

Chapters in Books:

[11]. Badri Nath Kapoor, "Practical Hindi-English Dictionary", (Prabhat Prakashan), 2012.