Volume 5, Issue 10, October-2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

# Di-phone-Based Concatenative Speech Synthesis System for Hindi

[1]Sangramsing Kayte, [2]Monica Mundada, [3]Dr. Charansing Kayte

[1, 2] Department of Computer Science & Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India

[3] Assistant Professor of Digital and Cyber Forensic, Aurangabad, Maharashtra, India

***Abstract— This Research paper describes the first Text-to-Speech (TTS) system for the Hindi language, using the general speech synthesis architecture of Festival. The TTS is based on diphone concatenative synthesis, applying TD-PSOLA technique. The conversion process from input text into acoustic waveform is performed in a number of steps consisting of functional components. Procedures and functions for the steps and their components are discussed in detail. Finally, the quality of synthesized speech is assessed in terms of satisfactoriness and articulacy.***

***Keywords— Di-phone concatenation, Speech synthesis.***

## I. INTRODUCTION

Text-to-speech synthesis enables automatic conversion of a sequence of type-written words into their spoken form. This paper deals with text-to-speech synthesis of Hindi language. A few attempts have been made in the past to cover different aspects of a possible TTS system for Hindi language [1][2]. However, no-one has succeeded in building a complete system providing high quality synthesized speech. We have worked on text-to-speech synthesis for a year. Here we describe the current version of our Hindi TTS system. Our improvements in the future will be based on this system.The synthesis task is performed here through the following two steps: analyzing text and producing speech. Each of these steps includes several modules, operating sequentially, as shown in Fig.1 [2]. At the first step, input text is normalized in the Text processing module. The tasks of this module cover sentence tokenization, non-standard words and homograph disambiguation. In the phonetic analysis module, letter-to-sound rules are used for finding pronunciations of the normalized words. Their intonation, which includes accent, boundaries, duration and F0 is produced in the Prosodic analysis module. At the second step, the synthesizer creates a speech waveform from the complete phonetic and prosodic description. The diphone concatenative synthesis is used to generate a waveform from a sequence of phones by selecting and concatenating units from a prerecorded database of diphones. Modifying the pitch and duration to meet the prosodic requirements is performed by TD-PSOLA technique [3]. At the end of the paper, results of evaluation of the system are given and discussed and some promising directions for Diphone synthesis techniques require less database as compared to the unit selection synthesis. It uses two adjacent phones to make the speech waveform. But this techniques suffers through the problem of coarticulation. Diphone synthesis is one of the most popular methods used for creating a synthetic voice from recordings or samples of a particular person; it can capture a good deal of the acoustic quality of an individual, within some limits. The rationale for using a diphone, which is two adjacent half-phones, is that the ``center" of a phonetic realization is the most stable region, whereas the transition from one ``segment" to another contains the most interesting phenomena, and thus the hardest to model.

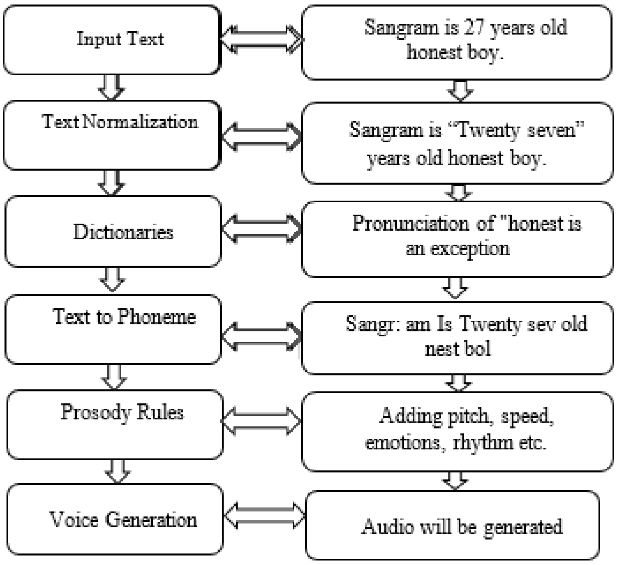
Input Text
Sangram is 27 years old honest boy.
Text Normalization
Sangram is "Twenty seven" years old honest boy.
Dictionaries
Pronunciation of "honest is an exception
Text to Phoneme
Sangr: am Is Twenty sev old nest bol
Prosody Rules
Adding pitch, speed, emotions, rhythm etc.
Voice Generation
Audio will be generated

Figure 1: General system architecture for Hindi TTS system

The diphone, then, cuts the units at the points of relative stability, rather than at the volatile phone-phone transition, where so-called co-articulatory effects appear. There is clearly a simplifying assumption: that all relevant phonetic realizations can be enumerated, and that by simply collecting all of phone-phone transitions, that any possible sequence of speech sounds in the target language could be produced. Thus, with a 44-phone inventory, one could collect a 44 * 44 = 1936 diphone inventory and create a synthesizer that could speak anything, given the imposition of appropriate prosody - intonation, duration, and shift in spectral quality, as determined by other modules in a general-purpose synthesizer.

## II. THE OFFICIAL LANGUAGE OF THE HINDI

The Indian constitution, in 1950, declared Hindi in Devanagari script to be the official language of the union. Unless Parliament decided otherwise, the use of English for official purposes was to cease 15 years after the constitution came into effect, i.e., on 26 January 1965. The prospect of the changeover, however, led to much alarm in the non-Hindi-speaking areas of India, especially Dravidian-speaking states whose languages were not related to Hindi at all. As a result, Parliament enacted the Official Languages Act, 1963,[4] which provided for the continued use of English for official purposes along with Hindi, even after 1965. In late 1964, an attempt was made to expressly provide for an end to the use of English, but it was met with protests from states such as Maharashtra, Tamil Nadu, Punjab, West Bengal, Karnataka, Puducherry and Andhra Pradesh. Some of these protests also turned violent. As a result, the proposal was dropped,[5] and the Act itself was amended in 1967 to provide that the use of English would not be ended until a resolution to that effect was passed by the legislature of every state that had not adopted Hindi as its official language, and by each house of the Indian Parliament. The current position is thus that the Union government continues to use English in addition to Hindi for its official purposes[6] as a "subsidiary official language,"[7] but is also required to prepare and execute a programme to progressively increase its use of Hindi.[8] The exact extent to which, and the areas in which, the Union government uses Hindi and English, respectively, is determined by the provisions of the Constitution, the Official Languages Act, 1963, the Official Languages Rules, 1976, and statutory instruments made by the Department of Official Language under these laws.

## III. HINDI TTS IMPLEMENTATION

The task of speech synthesis is to map a text to a waveform. The speech synthesis system performs this process in two steps:

1. Analyzing text: Converting the input text into a phonemic internal representation.
2. Producing speech: Converting the internal representation into a waveform.

The architecture of the system has a layered structure and each layer consists of functional components [9]. All required procedures and functions for the layers and their components will be defined in detail in the next sections.

### *A. Analysing Text*

In this step, the input text is analyzed and a phonemic internal representation is generated. Three modules (text processing, phonetic analysis and prosodic analysis) are used sequentially to carry out this step. These modules are described below.

### *B. Text processing*

The text processing module performs sentence tokenization, handles the nonstandard words and carries out homograph disambiguation. In Hindi, like English, whitespace (space, tab, newline, and carriage return) and punctuation can be separated from the tokens in the text. Each identified token is mapped to words, standard or non-standard. On-standard words are tokens like numbers or abbreviations, which need to be expanded into sequences of Hindi words before they are marked. These non-standard words are often very ambiguous. Dealing with non-standard words requires three steps: tokenization to separate out and identify potential non-standard words, classification to label them with a type from a predefined table, and expansion to convert each type into a string of standard words. There are a few homographs in Hindi. We need to disambiguate these homographs. Because knowledge of part-of-speech is sufficient to disambiguate some homographs, we store distinct pronunciations for these homographs labelled by part-of-speech, and then run a part-of-speech tagger to choose the pronunciation for a given homograph in context.

### *C. Phonetic analysis*

The phonetic analysis module takes the normalized word strings from the text processing module and produces a pronunciation for each word. The pronunciation is provided not just as a list of phones, but also a syllabic structure and lexical stress. The method for finding the pronunciation of a word is either by a lexicon or by letter to sound rules. For Hindi language, there exists a well-defined mapping from the orthography to the pronunciation, so lexicons for pronunciation are mostly redundant. Letter to sound rules do the whole job for almost all words in the Hindi language.

For well-defined languages like Hindi, writing rules by hand is simpler than training. We built hand-written rule sets for this system. Hand written letter to sound rules are context dependent re-write rules which are applied in sequence mapping strings of letters to strings of phones.

### *D. Prosodic analysis*

First, this module computes an abstract representation of the prosodic phrasing and pitch accent / boundaries of the text.Next, F0 values and duration are predicted from these prosodic structures. An utterance has a prosodic phrase structure in a similar way to it having a syntactic phrase structure [10]. Simple rules based on punctuation are a very good predictor of prosodic phrase boundaries for Hindi. We used the CART tree for predicting prosodic phrasing. This tree makes decisions based on distance from punctuation and whether the current word is a function word or content word. Intonation is provided by a CART tree predicting ToBI accents and an F0 contour generated from a model trained from natural speech. It used linear regression to assign target values to each syllable. For each syllable with a pitch accent or boundary tone, they predicted three target values, at the beginning, middle, and end of the syllable. They trained three separate linear regression models, one for each of the three positions in the syllable. Phones vary quite a bit in duration. Some of the duration is inherent to the identity of the phone itself. However, phone duration is also affected by a wide variety of contextual factors [10]. A speech database was constructed to study the duration model of the Hindi language. This database covers all the Hindi phonemes and their most frequent contextual combinations. It contains words of various syllabic structures in various locations. For predicting duration, the system used a method which employs a tree to predict zscores that is the number of standard deviations from the mean. These zscores are used to calculate segmental duration following the formula below [11]:

$$\text{Duration} = \text{mean} + (\text{zscore} * \text{standard deviation}) \qquad (12)$$

The first process was to provide means and standard deviations for each phone in the Hindi phoneset. The next process was to extract the features for predicting the durations. These features covered phonetic context, syllable, word position and type.

### *E. Producing Speech*

At this step, speech waveform is created from a complete phonetic and prosodic description consisting of a list of phones associated with duration and a set of F0 targets. Diphone concatenative synthesis is used for creating waveforms in the system. The diphone concatenative synthesis model generates a waveform from a sequence of phones by selecting and concatenating diphones from a prerecorded database of diphones. A diphone is a phone-like unit going from roughly the middle of one phone to the middle of the following phone. In this section, we will describe building a diphone database and diphone concatenation.

### *F. Building Diphone database*

First, we create an inventory of diphones for our system. Hindi has 15 vowel and 33 consonant phonemes, so there are 492 =1821 hypothetically possible diphone combinations. Not all of these diphones actually occur. In addition, our system does not bother storing diphones if there is no possible coarticulation between the phones, such as across the silence between successive voiceless stops. Thus the system has only 1044. Next we recruited speakers who had some vocal talent. The current system has two voices, one male and one female. Then a text was created for the speakers to say, and record each diphone. In order to keep recording diphones as consistently as possible, each diphone was recorded to enclose in a carrier phrase. It was an unaccented nonsense word, pronounced with a steady intonation. By putting the diphone in the middle of other phones, we keep utterance-final lengthening or initial phone effects from making any diphone louder or quieter than the others. We used different carrier phrases for consonant-vowel, vowel-consonant, phone-silence, and silence-phone sequences. Speech signals were recorded by a close talking microphone using a sampling rate of 16 kHz and 16 bit linear A/D conversion. After recording the speech, we labelled and segmented the two phones that make up each diphone. This was done automatically. But it was not completely accurate at finding phone boundaries, and so automatic phone segmentation was hand-corrected. Finally, pitch markers were manually set for voiced parts of the corresponding speech signal.

### *G. Diphone concatenation*

Given two diphones, in order to concatenate them, if the waveforms of the two diphones' edges across the juncture are very different, a perceptible click will result. Thus we applied a windowing function to the edge of both diphones so that the samples at the juncture have low or zero amplitude. Furthermore, if both diphones are voiced, the two diphones are joined pitch-synchronously. This means that the pitch periods at the end of the first diphone must line up with the pitch periods at the beginning of the second diphone; otherwise the resulting single irregular pitch period at the juncture is perceptible as well. Now, given our sequence of concatenated diphones, in order to modify the pitch and duration to meet our prosodic requirements, we used TD-PSOLA (Time-Domain Pitch-Synchronous Overlap-and-Add) for the process. Given an epoch-labeled corpus, the intuition of TD-PSOLA is that we can modify the pitch and duration of a waveform by extracting a frame for each pitch period, windowing the frame with a Hanning window, and then recombining these frames in various ways by simply overlapping and adding the windows pitch period frames. For assigning a specific duration to a diphone, to lengthen a signal with TD-PSOLA, we simply inserted extra copies of some of the pitch-synchronous frames, essentially duplicating a piece of the signal. For changing the F0 value of a recorded diphone, to increase the F0, we extracted each pitch-synchronous frame from the original recorded diphone signal, placed the frames closer together, with the amount of overlap determined by the desired period and hence frequency, and then added up the overlapping signals to produce the final signal. However, note that by moving all the frames closer together, we made the signal shorter in time. Thus to change the pitch while holding the duration constant, we added in duplicate frames.

## IV. EVALUATION

Some results of the performance assessment of the Hindi TTS are given. The adequacy of the system was tested in two ways: in terms of acceptability and of fluency. The synthesis output was directed to a Sound Blaster audio card. The experiment was performed with 10 subjects aged between 20 and 25 years, half of them female. They were asked to reviewer 50-100 words, phrases, sentences, and real texts all harvested from internet sources such as newspapers, literature magazines and publications. In our first experiment, fluency of synthesized speech was evaluated on two levels: word level and sentence level. Subjects, participating in the test were asked to write down everything they heard. Fig. 2 gives the percentage of correctly understood words and sentences, with word intelligibility rate being close to 86 %. In our second experiment, degree of acceptability of the synthesized speech was assessed, again on word and sentence level.

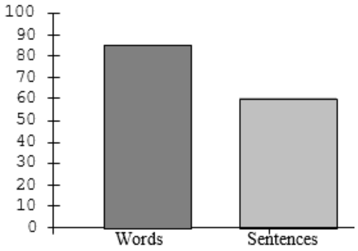

Figure 2: Intelligibility test. Percentage of correctly understood words and sentences.

Subjects were asked a few questions about naturalness, speed and sound quality and asked to mark how well the voice performs. The results are shown in Fig. 3, 4 and 5. Regarding the question whether the voice is nice to listen to or not, 37% considered the voice natural, 40% thought that the naturalness of the voice was acceptable and 23 % considered the voice unnatural.

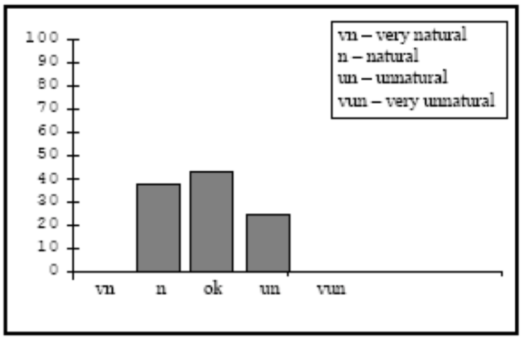

Figure 3: Naturalness of the voice

The speed of a system is a major concern, if the system speaks too fast or too slow this may have a negative effect on the concentration of the subjects. 76% of the listeners considered that the synthesized voice speed was normal rate.

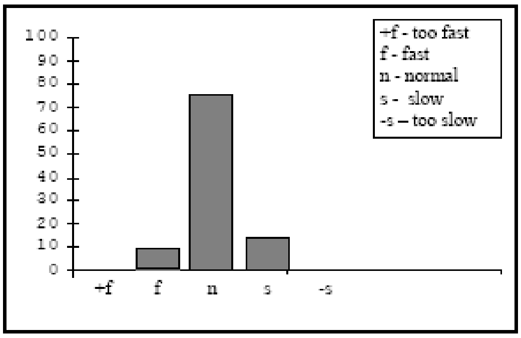

Figure 4: Speed of the speech

The question for this part was “Did you consider the synthesized voice has good sound quality?” 30% considered the voice has good quality; 60% thought the sound quality of the voice was neither bad nor good and the remaining 10% considered that the sound quality of the system bad.

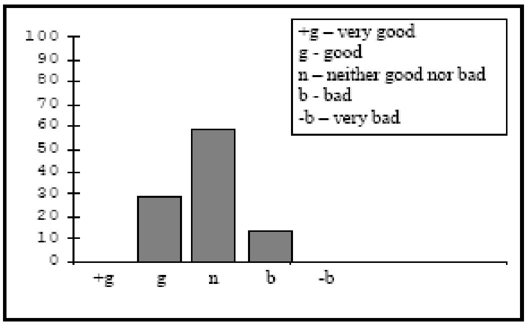

Figure 5: Sound quality of the voice

## V. CONCLUSION

The described speech synthesis system will be the first complete TTS system for the Hindi language. The earliest results are emerging. The synthetic speech produced by the current version of the system is intelligible, but utterances sometimes suffer from a lack of naturalness and fluency. Development of fluency and naturalness depends in particular on proper lexical stress assignment and a more sophisticated generation of prosodic features. The first attempts at developing a diphone-based synthesis system for the Hindi language are capable, so that further work on improving individual parts of the system is refreshed. At this stage, we are working on research to improve on the simple models to create more stylish trainable models.

## REFERENCES

[1] Sangramsing Kayte, Monica Mundada "Study of Marathi Phones for Synthesis of Marathi Speech from Text" International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-4, Issue-10) October 2015

[2] Sangramsing Kayte, Dr. Bharti Gawali “Marathi Speech Synthesis: A review” International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 3 Issue: 6 3708 – 3711

[3] Vivek Vijay Nar,Alice N. Cheeran, Souvik Banerjee “Verification of TD-PSOLA for Implementing Voice Modification” International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.comVol. 3, Issue 3, May-Jun 2013, pp.461-465

[4] "The Official Languages (Use for Official Purpose of the Union) - Rules 1976 (As Amended, 1987)". Archived from the original on 25 March 2010. Retrieved 10 June 2015.

[5] Forrester, Duncan B. (Spring–Summer 1966), "The Madras Anti-Hindi Agitation, 1965: Political Protest and its Effects on Language Policy in India", Pacific Affairs 39 (1/2): 19–36, doi:10.2307/2755179

[6] The Official Languages (Use for Official Purpose of the Union) - Rules 1976 (As Amended, 1987) - Paragraph 3(1)". Archived from the original on 25 March 2010. Retrieved 10 June 2015.

[7] Notification No. 2/8/60-O.L. (Ministry of Home Affairs), dated 27 April 1960

[8] The Official Languages (Use for Official Purpose of the Union) - Rules 1976 (As Amended, 1987) - Paragraph 1". Archived from the original on 25 March 2010. Retrieved 10 June 2015.

[9] Monica Mundada, Bharti Gawali, Sangramsing Kayte "Recognition and classification of speech and its related fluency disorders" International Journal of Computer Science and Information Technologies (IJCSIT)

[10] Black A., Taylor P., “The Festival Speech Synthesis System”, University of Edinburgh, Edinburgh, 2000 Available at http://www.cstr.ed.ac.uk/projects/festival

[11] Black A., Lenzo K., “Build Synthetic Voices”, Carnegie Mellon University, Edinburgh, 1999-2003

[12] Monica Mundada, Sangramsing Kayte, Dr. Bharti Gawali "Classification of Fluent and Dysfluent Speech Using KNN Classifier" International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 9, September 2014