ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(27), DOI: 10.17485/ijst/2015/v8i27/81721, October 2015* ISSN (Online) : 0974-5645

On Speech Synthesis of Sindhi Numeric

B. S. Ratanpal[1]* and Som Sahni[2]

[1]Department of Applied Mathematics, Faculty of Technology and Engineering, The Maharaja Sayajirao University of Baroda, Vadodara - 390 001, Gujarat, India; bharatratanpal@yahoo.com
[2]Department of Mathematics, ITM Universe, Dhanora Tank Road, Off Halol Highway, Paldi, Vadodara - 390 510, Gujarat, India; somsahni@gmail.com

Abstract

In a text-to-speech system, spoken utterances are automatically produced from text. The interest in text to speech synthesis increased in the world text to speech have been developed for many popular languages such as English, Devnagri, Punjabi and many other languages in the globe. Even today there are many researches and developments have been applied to these languages. In this paper the first attempt is made towards the development of text to speech of Sindhi numeric. Using the Java Sound API a sequence of Java programs are written for building text to speech synthesiser for Sindhi numeric. It is observed that generated output is accurate upto the significant level. Even the first version of this system is able to generate the synthesised utterance of words for inputted Sindhi numeric.

Keywords: Concatenation, Java Sound API, Recursive Logic, Speech Synthesis

1. Introduction

Among people there are some that are unable to read, either because of blindness (complete or partial), or for other reasons. There are also people who can read and write, but cannot speak. Therefore, initially for this category of people, there is a need for computer generated speech, namely the conversion of written texts into acoustic files.

Speech synthesis plays the important role in generating natural speech. There are several problems in pre-processing the speech synthesis, like acronyms, numbers, dates and abbreviations. The common problem in all these is converting the non standard words to the standard words. For example in Sindhi 353 should be pronounced as Te Sau Te Vanja and time 11:11 am should be pronounced as Subuh ja yarha lagi yarha minta. Several authors have worked on abbreviations and acronyms for example Chang J T, Schtze H and Alman R B[1] extracted abbreviation and acronyms from the text. Sproat R[2] extracted numbers from text. The text normalization problems have been carried out by several authors[3–7].

While generating the speech using synthesiser it is also important to consider the emotions like anger, love, melody etc. There are various methods one can opt for speech synthesis like, sentence tokenization and token translations, festival framework and Java Speech API. Each and every method has its own advantages and a disadvantage like festival framework is very powerful tool for speech synthesis but it requires tedious configuration and expert rules. The festival framework generates phonetic sound from existing database of speech sound. The generated sound is more robotic. On the other hand JAVA Speech API defines a cross-platform software interface to the speech synthesis. The Java Speech API, developed through an open development process with an extension of Java platform. These extensions are packages of classes written in Java programming Language. The developers can use functionality of the core part of the Java platform[9]. The Java speech API have been used by

* *Author for correspondence*

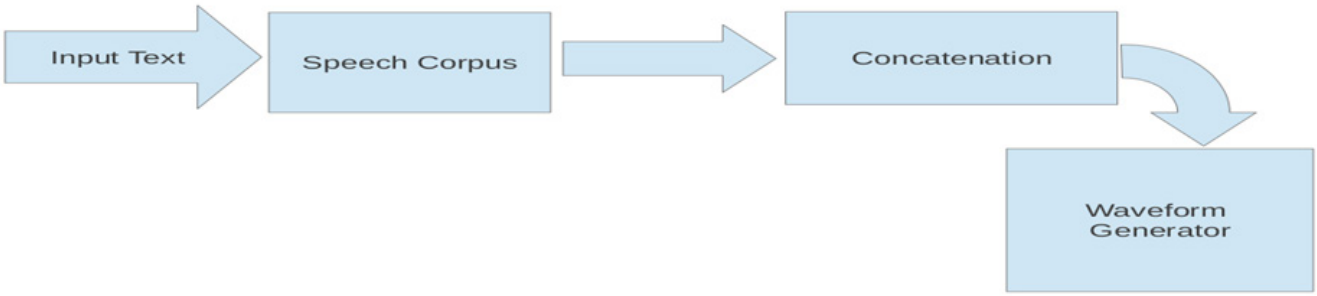

Figure 1. Basic structure of TTS system .

Sahani, Ramamohan and Pradhan[8] for the synthesis of Gujarati numeric.

In this paper we have implemented text to speech synthesiser for Sindhi numeric into words using Java Speech API. Section-2 contains method for implementation and section-3 contains the conclusion.

2. Text to Speech Synthesis

Speech synthesis involves algorithmically converting an input text into speech waveforms. Speech synthesizers can be characterised by the size of the speech units they concatenate, as well as the methods adopted to generate the speech. The choice of synthesis method is influenced by the size of vocabulary. There are two kinds of TTS system. Limited domain and Unlimited domain system. The system which can generate Speech for any input text is called Unlimited domain TTS. We have developed the limited domain TTS which can generate the speech for the input numeric.

2.1 Speech Corpus

The first step in developing the system is the recording itself. The principle of a concatenation-based speech synthesis is to concatenate speech segments from a speech segment database so that the synthetic speech mimics the voice of a speaker who recorded the speech corpus. So it is good to choose a professional speaker with a pleasant voice, good voice quality and possibly time-invariant speech quality. We have recorded all the common numeric in continuation including few names like sau, hazar, lakha, crore, arab.

The Sindhi numbers are recorded in .wav file format. Manually, we have segmented the combined .wav file for each numeric value including also few standard words of high place value such as hundred, thousand etc. The individual sound files are concatenated into a single sound file by calculating the length of each file and joining the next sound file, so that, there is no pause / space between the previous and next file. Java program is executed to con-concatenate these sound files for individual numeric digits as a single file without pause / silence. This wave file is loaded while the user enters the digit to generate the speech for the input number in Sindhi. Java program is written with recursive logic and during its execution it generates the Sindhi word sound corresponding to the input number in English numeric digit, using Java Speech API. The first step in developing the system is the recording itself. The principle of a concatenation-based speech synthesis is to concatenate speech segments from a speech segment database so that the synthetic speech mimics the voice of a speaker who recorded the speech corpus. So it is good to choose a professional speaker with a pleasant voice, good voice quality and possibly time-invariant speech quality. We have recorded all the common numeric in continuation including few names like sau, hazar, lakha, crore, arab.

The Sindhi numbers are recorded in .wav file format. Manually, we have segmented the combined .wav file for each numeric value including also few standard words of high place value such as hundred, thousand etc. The individual sound files are concatenated into a single sound file by calculating the length of each file and joining the next sound file, so that, there is no pause / space between the previous and next file. Java program is executed to concatenate these sound files for individual numeric digits as a single file without pause / silence. This wave file is loaded while the user enters the digit to generate the speech for the input number in Sindhi. Java program is written with recursive logic and during its execution it generates the Sindhi word sound corresponding to the input number in English numeric digit, using Java Speech API.

3. Conclusion

Currently there are no commonly used text to speech systems available for Sindhi language. This is first attempt to Speech Synthesis for Sindhi language. The generated output is accurate upto the significant level. Further, generated sound can be more realistic using mathematical techniques like Fourier transform Wavelet transform and filters.

4. References

1. Chang JT, Schtze H, Altman RB. Creating an online dictionary of abbrevations from MEDLINE. JAMIA; 2002.
2. Sproat R. Lightly supervised learning of text normalization: Russian Number Names. IEEE Workshop on Spoken Language Technology; Berkeley, CA; 2010.
3. Aw, et al. A phrase-based statistical model for SMS text normalization. ACL; 2006.
4. Pennell D, Liu Y. A character-level machine translation approach for normalization of SMS abbreviations. IJCNLP; 2011.
5. Nouman A, Jiang TY. Comparison of term frequency and document frequency based feature selection metrics in text categorization. Expert Systems with Applications. 2012; 39(5):4760–8.
6. Lei S, Xinming M, Lei X, Qiguo D, Jingying Z. Rough set and ensemble learning based semi-supervised algorithm for text classification. Expert Systems with Applications. 2011 May; 38(5):6300–6.
7. San-Segundo R, Montero JM, Giurgiu M, Muresan I, King S. Multilingual number transcription for text-to-speech conversion. 8th ISCA Speech Synthesis Workshop; Barcelona, Spain. 2013 Aug 31-Sept 2.
8. Sahani SN, Ramamohan S, Pradhan VH. On speech synthesis for Gujarati Numeric. Bulletin of the Marathwada Mathematical Society. 2013; 13(2):53–60.
9. Available from: http://java.sun.com/products/java-media/speech/forDevelopers/jsapi- guide/index.html