Nepali Text to Speech Synthesis System using FreeTTS

Krishna Bikram Shah[*], Kiran Kumar Chaudhary[+], Ashmita Ghimire[#]

[*+#]*Nepal Engineering College, Bhaktapur, Nepal*
[*]*krishnabikramshah@gmail.com*, [+]*kirankchaudhary11@gmail.com*, [#]*ashmitaghimire1@gmail.com*
[*+#]*Department of Computer Science and Engineering, Pokhara University, Nepal*

Krishna Bikram Shah *is Assistant Professor at Department of Computer Science and Engineering, Nepal Engineering College. He holds M. Tech. in Computer Science and Engineering from Sikkim Manipal University, India. He has shorter stays at several computing departments in India. His research interest include Artificial Intelligence, Bioinformatics and Machine Learning.*

Kiran Kumar Chaudhary *is a student of BE Computer Engineering, Batch 2011 at Nepal Engineering College and has received his bachelor degree from Pokhara University. He is currently working as a freelancer and also involved as an IT consultant with various companies.*

Ashmita Ghimire *is a student of BE Computer Engineering, Batch 2011 at Nepal Engineering College and has received her bachelor degree from Pokhara University. She is currently working as IOS application developer in a prominent software company.*

Abstract

This paper confers the tools and methodology used in developing a Nepali Text to Speech Synthesis System using FreeTTS and is entirely developed in Java and uses FreeTTS synthesizer. Vocalized form of human communication is Speech. Here the Nepali Language is Synthetized based on formant approach and the use of one of the popular generic frameworks FreeTTS that is available in public domain for the development of a TTS system. The Text To Speech Architecture has been developed putting more emphasis on the Natural Language Processing (NLP) component rather than Digital Signal Processing (DSP) component. Nepali language being mostly used language in Nepal and some parts of India and abroad, a text-to-speech (TTS) synthesizer for this language will prove to be a convenient tool and communication technology (ICT) based system to aid to those majorities of people who are illiterate and also to those who are physical impairments like visually handicapped and vocally disabled persons. This ability to convert text to voice may reduce the dependency, frustration, and sense of helplessness of these people. The system can be extended to include more features such as emotions, improved tokenization, interactive options and the use of minimal database.

Keywords— FreeTTS, Nepali Text-to-Speech, ICT, Speech Synthesis.

I. Introduction

Speech is the primary form of communication for human beings. Speech Synthesis is an artificial production of the human speech that allows transformation of the string of phonetic and prosodic symbol into a synthetic speech signal. TTS is a process through which input text is analysed, processed, understood and then the text is rendered as digital audio and then "spoken".

All the TTS system takes the text in digital format, such as ASCII for English, Unicode for Nepali. The quality of result is a function of the quality of the text, as well as of the quality of the speech generation process itself. The first requirement of TTS system is intelligibility and the second one is naturalness. [1, 2]. Text to speech (TTS) synthesis is the automated transformation of a text into speech that sounds as close as possible, as a native speaker or the language reading the text. Most Text to Speech Systems can be categorized by the method that they use to translate phonemes into audible sound. Some of them are Pre-recorded, Formant, Concatenated, etc.

TABLE I
Comparison between different categories of TTS

Element	Pre-recorded	Concatenated	Formant
Resource requirement	Very large storage, Small memory	Large storage, Very large memory	Low storage, relatively small memory
Vocabulary	Limited	Unlimited	Unlimited
Voice quality	Natural, Most pleasant	Natural	Robotic, sometimes not appreciated by the user
Multiple featured voices	Need high storage in that case	Need high storage in that case	Can produce multiple featured voices without any major changes
Intelligibility	High	Highly Intelligible	High

The TTS system is becoming more interactive and helpful to the users, especially physically and visually impaired and illiterate masses, i.e. not everyone can read text when displayed on the screen or when printed this may be because the person is partially sighted, or because they are not literate and these people can be helped by generating speech rather than printing or displaying it, using TTS system to produce the speech for the given text [3, 4].

So, that the TTS synthesis has a great demand for Nepali language. Speech application are primarily to simplify and automate tasks for human to make it easier assisting them in resolving certain problem domain. TTS synthesis has a wide range of application in everyday life because no longer people want to sit and read data from the monitor. Since there is painstaking efforts to be taken, this involves strain to theirs eyes. And with the growing popularity of digital material available through internet straight to everyone's handheld devices, the significance of TTS system has grown rapidly.

II. Text to Speech Tools

Text-to-speech synthesis is the automated transformation of a text into speech that sounds, as closer as possible, as a native speaker of the language reading the text. All TTS systems take the text in digital format, such as ASCII for English, Unicode for Nepali [18]. It is possible to build a TTS system which will work in combination with Optical Character Recognition system so that the system can read from printed text; but this does not affect the point, the output of the OCR system will be finally coded in corresponding digital text that is served as input for the TTS system. It is considered that a complete Text-to-speech system for any language must be able to handle text written in its normal form in that language, using its standard script or orthography. Hence a TTS which will accept Romanised input to speak out is not considered as a true TTS system. Normally, a text contains many inputs besides ordinary words. A typical Nepali text may contain numbers in different contexts; amount of money or phone number, percentage, acronyms and so on. A full text-to-speech system should be able to handle all these kinds of inputs with reasonable tolerance [19].

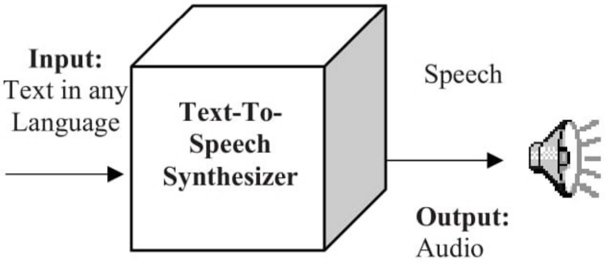

Fig. 1
Schematic Representation of text to speech (TTS) system

The earliest efforts of producing artificial sound were with different mechanical devices which model the human speech production system [6]. The earliest mechanical models had different music instrument like devices capable of producing only five long vowels each (a, e, i, o, and u). Hence the production of voice was limited to very basic vowels and the process of speech production was not automatic. These were the acoustic resonators modelling the human vocal tract. After a few generations other parts of the machine were improved; like pressure chamber for lungs, vibrating reed to act like vocal cords and leather tube for the vocal tract. With appropriate manipulation of the shape of the leather tube these machines were able to produce vowel sounds and consonants were simulated by separate constricted passages and controlled by the fingers. To simulate more sounds other components were added to the machine, e.g. movable lips, and tongue. Such machines were a good model of the human speech system but were limited to produce phones and limited set of words only, but were unable to produce long sentences.

First full electrical synthesis devices had a buzzer as excitation and resonant circuits to model the acoustic resonances of the vocal tract. These were able to generate single vowel sounds and no consonants. The first true speech synthesizer was introduced by Homer Dudley in 1939, called VODER (Voice Coder). The VODER consisted of wrist bar for selecting a voicing or noise source and foot pedal to control the fundamental frequency. Source signal was passed through ten bandpass filters whose output levels were controlled by fingers. Only skilled operator of VODER could produce a sentence of speech from the device. The demonstration of VODER showed that artificial production of speech was possible and increased more interest towards speech synthesis.

Further study on speech signal and its decomposition invented new technique of speech synthesis called formant synthesis with proper prediction of parameters representing the signal. Formant synthesis does not produce natural sounding speech when operated in fully automated mode to predict the signal parameters. With the advent of digital representation of digital sounds, availability of cheap and powerful computer hardware, and different digital signal processing techniques, speech generation method shifted from fully synthetic to concatenation of natural recorded speech. Speech generated from concatenation method resulted more closely to natural voice than the fully synthesized voice. In addition to this, since memory cost has been dramatically decreased and the processing speed has been exponentially increased the developers are nowadays interested in different concatenative approaches.

A. Text to Speech Systems for Indian and Nepali Languages

The first Nepali Text-to-Speech synthesizer that was built in 2000 by Sameer K. Maskey while he was doing his under graduation course in Carnegie Mellon University (CMU). The system was based on the Festvox tool built by Alan Black at CMU, US. A presentation of the software was also delivered at RONAST (Royal Nepal Academy of Science and Technology). Unfortunately, this first Nepali Text-to speech system appears to have been lost.

1) **Sambad :** The development of Nepali TTS at MPP began in 2005 with some preliminary reading about Festival and Festvox, and an introduction to speech in Sweden in September 2005. It started in earnest in Jan 2006 with a team of two software engineers (Srishtee Gurung and Ishwor Thapa). By the end of 2006 they had produced Nepali voices for both male and female implemented as a Web based application. This application can be used to read any Nepali text given as an input. In addition to it, in order to produce support for visually impaired and non-literate people the Nepali Voice has been integrated with the Ubuntu version of Linux to produce a Screen Reader, but the same is not available for Windows OS [20].

2) **Bharatsanchar NeLRaLEC:** Nepali TTS is being developed using the framework of Festival Speech Synthesis System developed by University of Edinburgh. This is free software which supports multi-lingual speech synthesis and has an open architecture for research in this field.

3) **Dhavni :** Indian Language Text-to-Speech System: In India, to help the visually impaired, vocally disabled and day to day increasing applications of speech synthesis has necessitated the development of more and more innovative text-to-speech (TTS) system. Some of the already developed TTS are described below. In this paper four Indian languages text-to-speech systems, namely Dhavni, Shruti, HP Lab system based on Festival framework.

III. Text to Speech Techniques

Naturalness and intelligibility are the two characteristics used to describe the quality of a speech synthesis system [8]. Naturalness of a speech synthesizer refers to how much the output sounds like the speech of a real person and the intelligibility of a speech synthesizer refers to how easily the output can be understood. The ideal speech synthesizer would have both qualities in it and each of the different synthesis technologies try to maximize these characteristics. There are few key technologies used for the generating synthetic speech waveforms: concatenative synthesis, formant synthesis, articulacy synthesis and some are discussed below.

A. Formant Synthesis

Formant synthesis [9] synthesize speech output using additive synthesis and an acoustic model taking parameters such as fundamental frequency, voicing and noise levels, which are varied over time to create a waveform of artificial speech. This technique is sometimes called rule based synthesis; though, many Concatenative systems also use rule based components. Formant synthesizers are usually smaller programs than Concatenative systems as they do not have a database of speech samples. This is also called the terminal analogy model. Numerous systems built on formant synthesis technology generate artificial, robotic-sounding speech and can be very reliably intelligible, even at very high speeds, avoiding the acoustic glitches that can often plague concatenative systems [3].

B. Concatenative Synthesis

Concatenative synthesis produces the most natural-sounding synthesized speech and is built on the concatenation of segments of the recorded speech, and possible modification of prosody such as modulation and duration [10]. Word level concatenation is not a viable since it requires recording of large level of units. Concatenative synthesis is probably the relaxed method to produce intelligible and natural sounding synthetic speech. Speech is synthesized by connecting pre-recorded natural utterances. However concatenative synthesizers are limited to one speaker and one voice and usually require massive memory than other methods and the significant aspect in concatenative synthesis is to find correct unit length. Longer units has high naturalness, contains less concatenation points and more control over the variation in phonemes (co-articulation) can be achieved. But the amount of required units and memory goes on increasing and with Shorter Units less memory is needed, but the sample collecting and labelling procedures become more difficult and complex. In contemporary systems units used are usually words, syllables, demi-syllables, phonemes, diphones, and sometimes even triphones.

C. Articulatory Synthesis

Articulatory Synthesis approach is based on computational models of the human vocal tract and the articulation processes occurring there. Few of these models are currently sufficiently advanced and are computationally efficient to be used in commercial speech synthesis systems. This method attempts to model the human vocal organs as perfectly as possible, so it is potentially the most satisfying method to produce high-quality synthetic speech [11] though it possess sufficient computational complexity as compared to other common methods.

IV. Implementation and Discussion

Formant synthesized speech can be very reliably comprehensible, even at very high speeds, avoiding the acoustic glitches that can often plague concatenative systems. Nepali TTS using FreeTTS has been developed based on this approach. The TTS system has text fed as input to generate synthetic speech as output for the corresponding text through audio device to create sound as described here under. After the text is fed to the TTS system the text pre-processing is done i.e. the input text is tokenized and each tokens is first matched and replaced with the English representation of the input Nepali text in the database dictionary. And for the words not found in the dictionary database, undergoes NLP. The text apart from the database is normalized, syllabified according to algorithm that are implemented and after that the syllables of the text is phonetically represented by the English text for each tokens and the tokens are passed to the DSP component to produce the sound of the words that is syllabalized and for other processing in the sound of the input text.

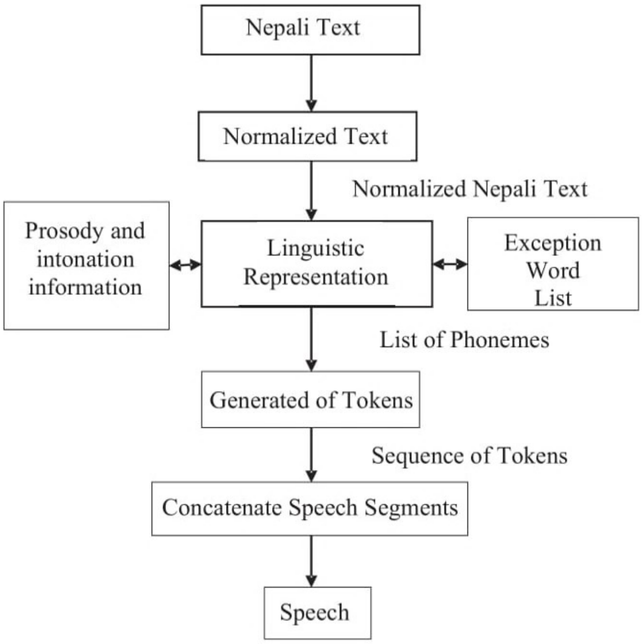

Fig. 2

Basic Flow Diagram of Nepali TTS synthesis System

A. Concatenative Synthesis

The text input will be in Nepali characters or basically in Unicode characters.eg. “मेरो देश नेपाल हो ।” The text is input in Nepali using keyboard that follows the Nepali words and text. The text processing module consists of preprocessing and syllabication.

B. Syllabication

In this approach, the syllabication algorithm breaks a word such that there are minimum numbers of breaks in the word, as minimum number of joins will have fewer artifacts. 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 along with syllable units, algorithm falls back on phones if bigger units are not available. A syllable types are: V, VC, CV, VCC, CVC, CCVC and CVCC etc. where V and C represent vowel and consonant respectively that are used for languages. There are twelve vowel found in Devanagari language the 12 Devanagari vowels. Devanagari script also has about 36 consonants.

Nepali phone set contains 11 vowels, 37 consonants and 21 diphthongs. We have stored these phone set into the speech dictionary to get pronounced by the system which are as below.

TABLE II

List of Nepali Phone Set

Vowel Alphabets (Swar Barna)	अ, आ, इ, ई, उ, ऊ, ए, ऐ, ओ, औ, अं,
Consonant Alphabets (Byanjan Barna)	क, ख, ग, घ, ङ, च, छ, ज, झ, ञ, ट, ठ, ड, ढ, ण, त, थ, द, ध, न, प, फ, ब, भ, म, य, र, ल, व, श, ष , स, ह, क्ष, त्र, ज्ञ
Half consonants	क्, ख् ग्, घ्, ङ्, च्, छ्, ज्, झ्, ट्, ठ्, ड्, त्, थ्, द्, ध्, न्, प्, फ्, भ्, म्, य्, र्, ल्, व्, त्र्

The system takes the Devanagari input and tokenizes each word in the input. Then, the tokens are checked and replaced in the database. And if not found in the dictionary, the text is goes NLP process where it is normalized i.e. it converts the non-standard word into standard words as the sentence may contain the abbreviation, numbers, and it needs to be converted into the words done by the normalization process. It then undergoes for syllabification (i.e. no of vowel with half consonant used in the formation of word like क्˙ आ˙ म्˙ अ . काम. Here c and cf is a vowel with half consonant s\ and d\ to form a word sfd) and phonetically equivalent English representation of Nepali phonemes is done (i.e. काम = kaama) for each tokens. Then those sequence of tokens are put into the FreeTTS[19] speech engine from where the speech output is generated

Algorithm 1: For Tokenization Process,

```
Input: Nepali Text Input (Si)
Output: Tokens (Ti)
Begin,
Step 1: Parse all text (Si) with the whitespace in the input text; where i = 1, 2, 3,.., n.
Step 2: For each input text (Si)
Extract words Wi = Si ; where i= 1,2,3,…, n.
Apply extract process for all text words i = 1,2,3,..,n.
Step 3: For each extracted words in Step 2,
Store Si[] = Wi; where i = 1,2,3,…,n.
Apply store for all words.
```

```
Step 4: Ti = Si;
Return (Ti).
Step 5: Pass value Ti for further processing.
End.
```

Algorithm 2.For Word Map from Dictionary,

```
Input: Tokens W = Ti, where i = 1,2,3,…n.
Output: English Words (We).
Begin,
Step 1: For every word W ← Word from Tokenization;
Length of the word (strlen);
Step 2: For W ε D
Select_Word ← Dictionary D;
Step 3: If
W = Select_Word
Return Word [1];
Else
Return W;
End.
```

Algorithm 3. For Natural Language Processing,

```
Input: Token W from Word Map check
Output: English Word (We)
Begin,
Step 1: For Every Word W
Check whether if the Word (W) is Numeric or Abbreviation,
Get the values of Numeric Characters or the Full Form for W.
Return Replaced_Word (W).
Step 2: For Every Replaced_Word (W)
Construct the Word, according to the Phone Set and Grammar.
Represent the word with syllables;
Return Syllabified_Word (W).
Step 3: For Every Syllabified_Word (W)
Map the Syllables with its Phonetically Equivalent Characters
Return Phonetic_Word(W).
End.
```

V. Analysis and Evaluation

Free TTS is a speech synthesis system written entirely in Java and is based upon Flite (a small run-time speech synthesis engine developed at Carnegie Mellon University). Flite is derived from Festival Speech Synthesis System from University of Edinburgh and FestVox project from Carnegie Mellon University. Free TTS now has the ability to import voice data from FestVox (US English only). With this, user can record their own voice using the FestVox tools, and then turn the resulting data into a Free TTS voice. For the project Free TTS is used as the DSP component for the Development of the TTS system.

For Evaluation Listeners are asked to rate the speech quality of the system, usually synthesizing the same set of sentences. Typically subjects are asked to rate the naturalness of the synthesis on a scale of 1-3, where 1 is poor, 2 is average and 3 is natural or gives correct pronunciation. All results are summed, and a mean score between 1 and 3 is derived, which is meant to represent the oral naturalness rating for the system.

The evaluation is done on the basis of:

- Intelligibility (how much of the spoken word can be understood)
- Naturalness or Pronunciation (how close to human speech does the output of the TTS system sound)

Test has been done to know whether the system has natural sounding or not as well as does it, sound closer to the real sound as human speak or not.

A. Evaluation of Individual Character

The Nepali language has 36 consonant and 11 vowels and 10 number characters for which evaluation was done to tabulate as below.

TABLE III
Evaluation of numbers, vowel and consonant used in test

S.N	Categories	Perfect Sounding	Average Sounding	Poor sounding
1.	Number	०, १, २, ५, ६, ७, ८	३	४, ९
2.	Vowel	आ इ ई उ ऊ ए ऐ ओ औ	अ	अं
3.	Consonant	क ख ग च छ झ ट ड त थ द ध न प फ भ म य र ल व स ष श ह त्र	घ ढ ठ ण ज ब क्ष	ङ ञ ज्ञ

Here, the test has been done to determine the naturalness i.e. accurate pronunciation of the number in the TTS system. The work has been done by dividing the task into three categories i.e. perfect, average, poor sounding characters. According to the evaluation the numbers like ०, १, २, ५, ६, ७, ८ are pronounced perfectly, the average sounding number is ३ and poor sounding numbers includes ४, ९. After calculation the accuracy was obtained and the accuracy for the perfect sounding is 70%, average sounding is 10%, and poor sounding is 20%. In this test rating is done for each categories i.e. for perfect sounding rate is given as 3, for average rate is given 2 and for poor rate is given 1. From which the weight is calculated of each categories to obtain the overall performance of number in the TTS system and its overall performance accuracy is 83.33% for numeric character set.

The perfectly pronounced vowel are आ इ ई उ ऊ ए ऐ ओ औ and its accuracy is 81.81%, the average sounding vowel is अ and its accuracy is 9.09%, the poor sounding vowel is अं and its accuracy is 9.09. After calculating the weight according to the rating method the overall performance accuracy is obtained as 90.90%.

Further, individual character or consonant that is perfectly pronounced are क ख ग च छ झ ट ड त थ द ध न प फ भ म य र ल व स ष श ह त्र and its accuracy is 72.97%, average sounding consonants are घ ढ ठ ण ज ब क्ष and its accuracy is 18.91%, poor sounding consonants are ङ ञ ज्ञ and its accuracy is 8.10%. Then weight is calculated according to the rating and then overall performance accuracy is obtained which is 88.28%.

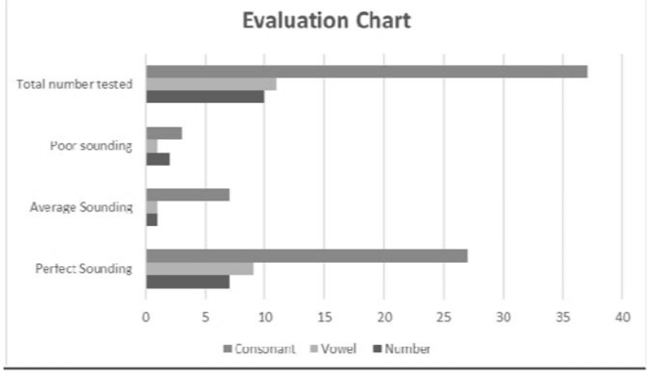

Fig: *Evaluation Chart for Vowel, Consonants and Numeric*

B. Evaluation of Individual Word

In the evaluation of the individual words the total words tested in the test were 2000 in which total natural sounded words are 1318, average sounded words are 257, and poor sounded words are 425. After analyzing the weight the overall performance were noted which is 81.56%.

TABLE IV
Evaluation Table for Individual keywords

Individual words	Perfect Sounding (3)	Average Sounding (2)	Poor Sounding (1)
Words used in test	1318	257	425
Accuracy of total words in percentile	65.9	12.85	21.25

Overall Performance

= (Sum of weighted words in all categories/Total Weighted words)*100

= (3954+514+425)/6000*100

= 81.5

Overall performance of the Nepali TTS system was found out to be 81.5% of the developed Nepali Text to Speech System

VI. Conclusions

This paper discussed the various steps involved in developing the Nepali text to speech synthesis system based on Formant Synthesis approach using FreeTTS. In the current system, linguistic feature such as intonation and prosody has not been implemented.

Further enhancement of the system could incorporating these features. However the system produces flat speech for any inputted Nepali text, though the synthesized speech doesn't sounds as natural as that spoken by a human. The system can be extended to include more features such as emotions, improved tokenization and use of minimal database.

References

[1] Dutoit T., "An Introduction to Text-To-Speech Synthesis"¸ Kluwer Academic Publishers, 1996.

[2] Allen J., Hunnicut S., Klatt D., "From Text To Speech, The MITTALK System", Cambridge University Press, 1987.

[3] J.Sangeetha, S.Jothilakshmi , S. Sindhuja , V. Ramalingam ,Text to Speech synthesis system for Tamil, International Conferenceon Information Systems and Computing (ICISC-2013),India.

[4] K. Kamble and R. Kagalkar, A Review:Translation of Text toSpeech Conversion for Hindi language, International Journal of Science and Research (IJSR) Volume 3 Issue 11, November,2014.

[5] Van Santen, Jan P. H.; Sproat, Richard W.; Olive, Joseph P.; Hirschberg, Julia (1997). Progress in Speech Synthesis. Springer. ISBN 0-387-94701-9.

[6] NepaliTTS.Online:http://www.bhashasanchar.org/pdfs/NepaliTTS_%20manual.pdf. Access Date: 14th October, 2015.

[7] Hariharan, R. [Online]. Available: http://dhvani.sourceforge.net/. Accessed on 20 February 2012.

[8] Speech Synthesis. Online: http://en.wikipedia/wiki/Speech_Synthesis. Access Date: 4th November, 2015.

[9] D. Klatt, "Software for a cascade/parallel formant synthesizer," Journal of the Acous-tical Society of America, vol. 67, pp. 971–995, 1980.

[10] D. Jurafsky and J.H. Martin, Speech and Language Processing. Pearson Education, 2000.

[11] P. Rubin, T. Baer and P. Mermelstein, "An articulatory synthesizer for perceptual research," Journal of the Acoustical Society of America, vol. 70, pp. 321–328, 1981.

[12] Black A., Taylor P., Caley R. (1999) The Festival Speech Synthesis System Documentation v1.4. (http://www.cstr.ed.ac.uk/projects/festival/manual/).

[13] Building Synthetic Voices, by Alan W Black and Kevin A. Lenzo, For FestVox 2.0 Edition 1999-2003 by Alan W Black & Kevin A. Lenzo

[14] Dutoit T., "An Introduction to Text-To-Speech Synthesis"¸ Kluwer Academic Publishers, 1997.

[15] D. Jurafsky and H. James, "Speech and language processing an introduction to natural language processing, computational linguistics, and speech," 2000.

[16] David Öhlin, Rolf Carlson "Data-driven formant synthesis" Proceedings, FONETIK 2004, Dept. of Linguistics, Stockholm University.

[17] A. Trilla, Natural Language Processing techniques in Text-To-Speech synthesis and Automatic Speech Recognition. Departament de Tecnologies Media Enginyeria i Arquitectura La Salle (Universitat Ramon Llull), Barcelona, Spain atrilla@salle.url.edu, 2009.

[18] Bhusan Chettri, Krishna Bikram Shah, Nepali text to speech synthesis system using ESNOLA method of concatenation, International journal of computer applications (0975 - 8887), vol. 62, no. 2, pp 24-28, January 2013.

[19] M.J. Liberman, K.W. Church, "Text Analysis and Word Pronunciation in Text-to-Speech Synthesis", Advances in Speech Signal Processing, S.Fumy, M.M. Sondhi eds, Dekker, New York, pp. 791-831, 1992.

[20] Gurung, Sristhtee and Ishwor Thapa. 2007. 'Building Nepali text-to-speech'. In Rai etal. (eds.) Recent studies in Nepalese linguistics.

* * *