IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 5, Ver. I (Sep.- Oct. 2017), PP 14-21
www.iosrjournals.org

Speech Synthesis for Punjabi Language Using Festival

*Pansy Nandwani[1], Dr. Dharam Veer Sharma[2]
[1](Department of Computer Science, Punjabi University, India)
[2](Department of Computer Science, Punjabi University, India)
Corresponding Author: Pansy Nandwani

Abstract: *Processing of digital speech plays an important role in modern speech communication research and applications. The main propose of digital speech is communication which means transmission of messages between human and computer systems. The process of converting the written text into speech is called Text-to-Speech (TTS) and the generation of "synthetic" speech is known as Speech Synthesis. This paper discusses the development of Text-to-Speech system using festival framework for Punjabi language which is in Gurumukhi script. Since festival uses concatenation approach, diphone has been chosen as the basic unit of concatenation for speech synthesis. We defined the phone set and possible number of diphones for Punjabi language. Apart from that configuration of Festival framework on Ubuntu 14.04 Linux is discussed. Recording of nonsense word has been done from which the diphones are extracted for concatenation to produce speech. The aim of this paper is to give the details regarding the manipulations of some files required to add some new language in festival.*

Keywords: *Concatenation Synthesis, Diphone, Festival, Punjabi, Speech synthesis.*

Date of Submission: 22-08-2017 Date of acceptance: 08-09-2017

I. Introduction

Text-to-Speech or Speech Synthesis systems has been the mainstream area of research in the field of Natural Language Processing (NLP). Natural Language Processing (NLP) is a way for computer systems to analyze, perceive, and deduce the meaning from human language (Natural Language) in useful and bright way. With the application of natural language processing, one can construct and shape the knowledge to perform various activities translation, Speech-to-Text conversion, relationship extraction, Text-to-Speech conversion, sentiment analysis, summarization, etc. Natural Language Understanding and Natural Language Generation are the two main components of natural language processing. The process of converting the written text into sound or speech is called as Speech Synthesis. Speech Synthesis it the reverse process of Speech Recognition (Speech-to-Text). The performance parameters for these Text-to-Speech systems are intelligibility, naturalness and speed. There are several speech synthesis systems like eSpeak, Chair, Mi-Talk and Festival which use different techniques that have been developed for converting the written text (input) to speech waveform. Dhvani, Shruti, Vani and so many speech synthesis systems that have been developed for Indian languages like Hindi, Tamil, Marathi etc. The motive is to build a speech synthesis system based on concatenative approach for Punjabi language. Festival system is a multi-lingual speech synthesis i.e. it is designed to support multiple languages. This system supports various languages like English, Welsh and Spanish and also supports some Indian languages (Hindi, Telugu, Marathi). The purpose is to develop TTS system which supports Punjabi language using the Festival framework.

II. Text-To-Speech

Speech plays a vital role as a medium of communication which human use to interact with each other. Digital speech is the medium for communication in between human and computer systems. The process of converting the written text into speech is called **Text-to-Speech (TTS)** and the generation of "synthetic" speech is known as **Speech Synthesis.** Text to speech enable the reading of computer display information for the visually challenged person, or may simply be used to intensify the reading of a text message. Text to speech system processes is very different from live human speech production. Live human speech production depends on fluid mechanics which further depends on changes in lungs' pressure and vocal tract contractions. The objective of a speech synthesis system is to convert any given text into a corresponding speech waveform.

Text processing and speech generation are the two phases of a text to speech system. The main motive of the text processing component is to process the given input text and produce befitting sequence of phonemic units. These phonemic units are accomplished by the speech generation component either by selection of a unit from a large speech corpus or by synthesis from statistical parameters [1].

2.1 Challenges in Speech Synthesis Systems

The important aspects of Speech Synthesis systems are intelligibility and naturalness that characterize these systems. Intelligibility is the ease with which output is understood and the naturalness describes how closely the output synthesized speed is like human speech. It is difficult to achieve both the qualities [2].

2.2 Techniques of Speech Synthesis

In order to generate speech there are three main categories of speech synthesis techniques.

2.1.1 Articulator Synthesis

The mechanical and acoustic model of speech production are used in articulator synthesis to generate synthesized speech. This synthesis produces intelligible synthetic speech, but the disadvantage of this system is output is still far from natural sound. In this synthesis process, the articulation system and vocal tract used by humans is modelled. In order to produce speech, human use various articulators like lips, teeth, mouth, tongue etc. Thus this articulation synthesis process is an artificial process to produce voice and is based on natural way of producing the voice. The computational complexity of this method is more as compared to the existing methods [3].

2.1.2 Formant Synthesis

A set of rules are defined in Format Synthesis which describes the way to modify various parameters such as pitch and frequencies. These rules are based on source-filter model of speech production. It is also known as **Synthesis by Rule** [4]. Prosody can be added to speech by modifying the filter parameters. Due to usage of formant synthesis robotic sound is generated. In order to generate speech an acoustic modeling is used. In comparison to concatenation based approach the sound produced is intelligible in nature.

2.1.3 Concatenative Synthesis

Instances of recorded sound units are concatenated to produce speech. User specific sequence of sound can be produced from database which is constructed from recording of other sequences. Units for concatenative synthesis could be phone which is a single unit of sound, di-phone which is also a unit of speech but made of two speech sounds (phones), words etc.

The size of the speech units stored influence the quality of synthesized speech, larger the size of speech unit (sentences) the more natural will be sound, but it will restrict the flexibility of the text-to-speech system. Whereas if we have small speech units (phones) in inventory it will provide more flexibility but the quality of speech will degrade. Hence selection of befitting unit is very essential [3][4].

Types of concatenative speech Concatenative speech synthesis can be broadly classified into three categories.

a) Unit Selection Synthesis

In unit selection synthesis, enormous databases of pre-recorded sounds are used. It is possible to synthesize more natural sound with the use of large databases which contains large number of units having different prosodic characteristics. During database creation, each recorded utterance is divided into phones, diphones, syllables, words, phrases, and sentences.

b) Diphone Synthesis

Diphone synthesis uses a minimal speech database which contains all the di-phones arising in a language. In this type of synthesis, only one sample of each diphone is present in the database. Speech produced from di-phone synthesis will be more natural than speech produced from format synthesis but quality of speech will be degraded than that of unit-selection synthesis.

In format based technique, the quality of the sound is good but not as natural as in unit selection synthesis. Due to use of digital signal processing the voice produced is less natural. It is widely used in commercial applications. In digital signal processing, MBROLA, Linear Predictive coding (LPC) are used. The limitation of this approach is that it there occurs sonic glitches.

c) Domain-Specific Synthesis

Domain-specific synthesis provides the way to concatenate the words and phrases that were previously recorded so as to generate complete utterances. It is applied in areas, where the output is limited to a particular domain such as like transit schedule announcements or weather reports [2].

III. Punjabi Language

Punjabi language is the member of Indo-Aryan languages. It is one of the language which is recognized by Indian constitution in 8th schedule. Punjabi language is spelled as Panjabi which is an official language of state Punjab. In India, Punjabi is written in Gurmukhi Script. The Gurmukhi script is written from left to right. In Pakistan, Punjabi is written in Shahmukhi script Punjabi. It is also the 10th commonly spoken language in world. Origination of Punjabi language is from Devanagari language. About 100 million people use Punjabi Language for oral communication. Around 33 million speakers of Punjabi language are in northern states of India that are Punjab, Himachal Pradesh, and Haryana and approximately 76 million speakers are there in Pakistan [5].

The main characteristic of Punjabi language is that it is tonal language. Consonants are differentiated by using various tones otherwise they would be identical. List of Punjabi consonants with properties place of articulation (POA) and manner of articulation is shown in Table 1. Punjabi language has 10 vowels These vowels can either be short or long. List of Punjabi vowels with its properties in Table 2.

IV. Festival

Festival provides a general framework for to develop Text-to-Speech systems. It provides full speech synthesis facilities through multiple Application Programming Interfaces and uses the Edinburgh Speech Tools Library. Festival is written in C++ and for control, it has Scheme (SIOD) based command interpreter. It is free software which is distributed under an X11-type license allowing unrestricted non-commercial as well commercial use. Festival is highly flexible and multilingual. Festival has modules for text processing, prosodic processing and waveform generation. This framework supports di-phone synthesis and unit selection synthesis. The units are stored in speech database based on prosodic and phonetic context. It uses a concatenation technique for generating speech waveform.

4.1 Requirements

A UNIX machine, A C++ compiler, GNU make, NCurses library, Festival Speech Synthesis System source, The Edinburgh Speech Tools Library, Festvox Package, Lexicon distribution. We presume that we already have Ubuntu 14.04. Following the commands to install GNU make and C++ compiler and NCurses library.

$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install build-essential
$ sudo apt-get install libncurses5-dev

4.2 Installation [6]

Firstly, download festival-2.4-release.tar, speech_tools-2.4-release.tar, festvox-2.7.0 release.tar, festlex_OALD.tar.gz, and festlex_POSLEX.tar.gz and any voice package, place them under one directory. Change your working directory to the directory where the packages have been placed. Then unpack the packages one by one. To extract or unpack tar.gz files use **tar -xzvf filename.tar.gz** command. For example,

tar -xzvf festival-2.4-release.tar.gz
tar -xzvf speech_tools-2.4-release.tar.gz
tar -xzvf festvox-2.7.0-relaese.tar.gz

Following are the commands to compile festival, Speech_tools and festvox.

4.2.1 Compiling speech_tools

To compile speech_tools, first change the directory to speech_tools with cd command.

$ cd speech_tools

Then run the following commands

$ make test
$ make install

Then get back to the directory where you have placed all the packages with the command cd ..

$ cd ..

4.2.2 Compiling Festival

To compile festival, change the directory to festival with the command cd (change directory)

$ cd festival

Then run the following commands

$ ./configure

Bunch of output lines will be seen after running the above command. Then run the following commands

$ make
It will take some time
$ make test
$ make install
Then get back to the directory where you have placed all the packages with the command cd ..
$ cd ..

Table 1 Punjabi Consonants

			Velar	Palatal	Retroflex	Dental	Bilabial	Labio-dental	Glottal
Stop	Voiced	Unasp	ਕ		ਟ	ਤ	ਪ		ਗ਼
		Asp	ਖ		ਠ	ਥ	ਫ		
	Voiceless	Unasp	ਗ		ਡ	ਦ	ਬ		
		Asp	ਘ		ਢ	ਧ	ਭ		ਖ਼
Affricative	Voiced	Unasp		ਚ					
		Asp		ਛ					
	Voiceless	Unasp		ਜ					
		Asp		ਝ					
Nasal					ਣ	ਨ	ਮ		
Fricative	Voiced					ਜ਼			
	Voiceless			ਸ਼		ਸ		ਫ਼	ਹ
Approximant				ਯ		ਲ		ਵ	
Flap					ੜ / ਲ਼	ਰ			

4.2.3 Compiling Festvox

In the same way as we did for festival and speech tools, change directory to festvox
$ cd festvox
$ ./configure
$ make

Table 2 Punjabi Vowels

	Back	Near-back	Central	Front	Near front
Open			ਆ		
Open-mid	ਔ			ਐ	
Mid			ਅ		
Close	ਊ			ਈ	
Close mid	ਓ	ਉ		ਏ	ਇ

Export the three environment variables named PATH, ESTDIR, and FESTVOXDIR we have to do it every time we run festival in a new terminal. Following are the three export commands that we need to execute to set up variables. As per the path of our directory.
$ export PATH=/home/pansy/fest/festival/bin:$PATH
$ export ESTDIR=/home/pansy/fest/speech_tools
$ export FESTVOXDIR=/home/pansy/fest/festvox
Now execute the following command to run festival
$ festival
Run the following command to check if it is working or not.
festival> (SayText "Hello Pansy Nandwani!")
Upon the successful installation of festival one can hear the voice "hello pansy Nandwani". To exit the festival, run
festival> (exit)

V. Implementation

Following are the steps to build the Speech Synthesis system for Punjabi language using festival.

5.1 Construct the Basic Template Files

To add voice in festival first of all create a directory to hold the voice.
$ mkdir net_pa_pn_diphone
Then change the directory to the new one that is net_pa_pn_diphone and build the basic structure of this directory be executing the following commands.
$ cd net_pa_pn_diphone
$ FESTVOXDIR/src/diphones/setup_diphone net pa pn
where net, pa, pn are arguments the arguments to setup_diphone, net is the institution name, pa is the two letter ISO standard for Punjabi language, and pn is the initials of the name of the speaker.

5.2 Define the Phoneset for Punjabi

The next step is to generate the phone-set definition. A phone-set is a set of symbols which is defined in terms of features, like whether it is consonant or vowel, what is the type of consonant, what is place of articulation for consonants, vowel frontness etc. In phone-set these features and their values must be defined according to table 1 and table 2 [6]. In net_pa_pn_phone.scm file phone-set can be defined as follows. Note that in festival scheme files comments are mentioned by putting the semi-colon in front of the line.

```
(defPhoneSet
Name of the phone-set

Feature definition
Phone definition)
```

The name of the phone-set in this case is net_pa. Feature definition is a list of definitions of features which consists of name of feature and its values. For example

```
(
        ;; vowel or consonant
         (vc + - 0)
        ;; vowel frontness: front mid back
          (vfront 1 2 3 0 -)
        ;; consonant type: stop affricative fricative nasal flap approximant
          (ctype s a f n l r 0)
        ;; place of articulation:labio-dental alveolar palatal bilabial retroflex velar glottal
          (cplace l a p b r v g 0)
        ……
)
```

The third section consist of definition of phones. Each phone is defined in terms of phone name and the values for each feature in the same order as the features are defined in above section. It is also necessary to include a definition for silence phones. In addition to the definition of the set the silence phone(s) themselves must also be identified to the system. This is done through the command PhoneSet.silences. For example,

```
(
        ; silence
    ( pau - - 0 0 0 0 0 0 - - - )
    ; vowels
    (  a - + s 2 2 - 0 0 0 - 0 ) ; ਅ
    (  aa - + l 3 2 - 0 0 0 - 0 ) ; ਆ
    ; consonants
    (   k - - 0 0 0 0 s v - - - ) ; ਕ
    (  kh - - 0 0 0 0 s v - + - ) ; ਖ
    )
(PhoneSet.silences '(pau)
```

5.3 Generate Diphone Schema File

For the clean articulation of diphones, one technique is to use the carrier sentences. Nonsense words are used as carrier sentences which ensure the consistent pronunciation of diphones. There are many benefits of taking the nonsense words as carrier words.

- We don't have to look for natural examples containing desired diphone.
- Presentation of nonsense words are less prone to errors.
- List of diphones can be checked easily.

The nonsense word formation has been done the way that kal-diphone (US-English) does. In this work**, V – SIL, SIL - V, C - C, C- V, V - V, V - C, SIL - V, SIL – C** pairs have been considered where c is meant for consonants, v is meant for vowel and SIL is meant for silence. The diphone generation algorithm will is as follows.

For V - C pair: The nonsense word is in the form of "pau taa taa VV pau"

```
for each vowel in list of vowels
    for each consonant in list of consonants
        diphone = taa+tV+Caa
```

Similarly,

For C - C pair: The nonsense word is in the form of "pau taa C-C aa taa pau"
For C - V pair: The nonsense word is in the form of "pau taa CV taa pau"
For V - V pair: The nonsense word is in the form of "pau taa tVV taa pau"
For SIL - V pair: The nonsense word is in the form of "pau V taa pau"
For SIL - C pair: The nonsense word is in the form of "pau Caa taa pau"
For V - SIL pair: The nonsense word is in the form of "pau taa aa V pau"
For C - SIL pair: The nonsense word is in the form of "pau taa aa C pau"

In our phoneset we considered 36 Consonants, 10 vowels, and 1 silence phoneme i.e. pau. Here is the calculation for Punjabi diphones.

CC = 36*36 = 1296 (1-1296)
CV = 36*10= 360 (1297-1656)
VC = 10*36 = 360 (1657-2016)
VV = 10*10 = 100 (2017-2116)
SIL-V = 1*10 = 10 (2117-2126)
SIL-C = 1*36 = 36 (2127-2162)
C-SIL = 36*1 = 36 (2163-2198)
V-SIL = 10*1= 10 (2198-2208)
SIL-SIL= 1*1= 1 (2209)

Write the code in pa_schema.scm file. Run the code with the following commands to generate all the possible diphones.

$ festival festvox/pa_schema.scm festvox/diphlist.scm.
festival> (diphone-gen-schema "pa" "etc/padiph.list")

After running these commands, the list of diphones will be generated in the following format.

(pa_0001 "pau t aa k - k aa t aa pau" ("k-k"))
(pa_0002 "pau t aa k - kh aa t aa pau" ("k-kh"))

Each line in padiph.list contains file id, a prompt, and a diphone name. The file id is used to in the filename for the waveform, label file, and any other parameters files associated with the nonsense word.

5.4 Recording and Labelling of Nonsense Words

If recording is to be done through Linux, then need of prompt setup is required. But we have done recording with audacity recorder. This recorder will record .wav files. Identify and note down the start and end of diphone in each nonsense and save them with the id of diphone (defined in padiph.list). Copy all the .wav files to /net_pa_pn_diphone/wav directory.

5.5 Generate the Index File

The next step is to create the index file or diphone index after labelling the nonsense words, this index file identifies which diphone comes from which files, and from where. The index file is the simple text file which consists of a simple header, followed by a single line for each diphone: diphone name, the file id, start time of diphone, mid-point of diphone and end time of diphone. Here is the example of diphone index file format.

EST_File index
DataType asci
NumEntries 2209
IndexName pn_diphone
EST_Header_End
k-k pa_0001 0.808 0.924 1.040
pau-pau pa_2209 1.725 1.9435 2.162

The number of entries field must be correct; if it is small than actual number of entries it will ignore the rest of entries after that point.

5.6 Extract Pitchmarks and LPC Coefficients

The pitchmarks can be extracted and the program to move the pitch marks to nearest peak is also provided with the following commands but before running these commands make the copy of padiph.list in /net_pa_pn_diphone/etc and "name it text.done.data"

$ bin/make_pm_wave wav/.wav*
$ bin/make_pm_fix pm/.pm*

Then run the following command to build the pitch-synchronous LPC coefficients

bin/make_lpc wav/.wav*

5.7 Test the Phone Synthesis

This is the stage where testing of wave synthesizer is being done. Lot of work is still pending but correction of labelling errors can be done now. To test the phone synthesis first start the festival then enter the string of phones in the following manner

$ festival festvox/net_pa_pn_diphone.scm "(voice_net_pa_pn_diphone)
festival> (SayPhones '(k s r))

It will speak diphones k-s and s-r.

5.8 Addition of Lexicon and Letter-to-Sound Rules

The pronunciation of a word can be defined either by a lexicon (a large list of words and their pronunciations) or letter to sound rules. In festival, pronunciation of a word is defined with not only list of phonemes but syllabic structure is also required. In lexicon structure of festival both word and part of speech is required, in addition to that a stress value is also given [7]. There is a need of large lexicon containing tens of thousands of entries that is used for standard implementation of voice. These lexicon entries are kept in text file named net_pa_lex.out which is created under /net_pa_pn_diphone/festvox (as per the convention). Example for the format of lexicon entry are is written below.

("ਹਰਮਨ" n (((h r) 0) ((m n) 0)))

After creating the lexicon file add the following code below the comment "lexicon definition" in net_pa_pn_lexion.scm.

(lex.create "net_pa")
(lex.set.phoneset "net_pa")
(if (probe_file (path-append net_pa_pn_dir "festvox/net_pa_lex.out"))
(lex.set.compile.file (path-append net_pa_pn_dir "festvox/net_pa_lex.out")))

But in case of when word is not explicitly mentioned in lexicon then Letter to sound rules (LTS) are used as backup [7] [8]. It is easy to write letter to sound rules by hand but it requires a lot of linguistic knowledge. Add these letter to sound rules in net_pa_pn_lexicon.scm file below the comment "Hand written letter to sound rules". Write letter to sound rules as per the following format.

(Left context [Middle item] Right context = New item)
For example
(lts.ruleset
net_pa

((Vowel a aa i i: u u: e aI o aU))
(
([ਅ] = a)
([ਆ] = aa)
…….
))

Note that Unicode must first be converted into three byte ASCII form in letter to sound rules.

5.9 Addition of Prosody

As this step deals with the intonation model and duration of phonemes, net_pa_pn_intonation.scm and net_pa_pn_durdata.scm files must be considered. There are various number of intonation models available but for this work simple intonation model has been used. For predicting the duration of phonemes various modules are available. There is an important parameter 'Duration_Stretch whose value is multiplied by the duration of every phoneme predicted by any model.

5.10 Testing and Evaluation of Voice

Now the basic synthesizer is ready although there is much to do. Now you van test the voice by executing the following commands.

$ festival festvox/cmu_ja_awb_diphone.scm "(voice_cmu_ja_awb_diphone)"
festival> (SayText " □□□□ □□□ ਹਰਮਨ□□ ")

VI. Conclusion and Fututre Scope

Speech Synthesis systems are applicable in various fields. These systems are used by many educational institutes, vocally impaired persons, web browsers, and many more companies which are dealing with Speech Synthesis and Speech Recognition integrated with cell-phones applications. Many systems have been developed for many language using different approaches, but here the emphasis is on Punjabi language using Diphone concatenation approach. Festival supports many Indian languages like Hindi, Tamil, Bangla etc. except Punjabi. Thus diphone concatenation has been used to make it work for Punjabi language also. Right now the phone-set for 36 consonants and 10 vowels have been described. Since, for production of speech diphones has been concatenated, due to the smaller speech units the quality of voice is degraded. To improve this of larger speech-units like syllable can be used, but it will also increase the size of database. Even to decrease size of diphone database non-existing diphones can be ignored. In Future, the phone-set can be extended for tippi, addhak, as well as bindi. Tokenizers can be extended to identify Non Standard words like numbers and abbreviations. After the development of full system, it can be integrated with many applications like Web Browsers and text editors.

Acknowledgement

The completion of any research depends upon cooperation, co-ordination and efforts of several sources of knowledge. I am grateful to Dr. Dharam Veer Sharma (Associate Professor, Department of Computer Science, Punjabi University, Patiala) for his willingness to give me direction and valuable advice. I am thankful to him for providing immense guidance.

References

[1] Mache Suhas R., Manasi R. Baheti, and C. Namrata Mahender, Review on Text-To-Speech Synthesizer, International Journal of Advanced Research in Computer and Communication Engineering,4(8), 2015, 54-59.
[2] Gupta Shruti and Parteek Kumar, Comparative study of text to speech system for Indian Language, International Journal of Advances in Computing and Information Technology, 1(2), 2012, 199-209.
[3] Mark Tatham and Katherine Morton, Developments in Speech Synthesis (United Kingdom, John Wiley & Sons, Ltd., 2005).
[4] Paul Taylor, Text-to-Speech Synthesis, (University of Cambridge, Cambridge University Press, 2009).
[5] Andrea Lynn Bowden, Punjabi Tonemics and the Gurmukhi Script: A Preliminary Study, Brigham Young University, United States of America, 2012.
[6] Roy Somnath, A Technical Guide to Concatenative Speech Synthesis for Hindi using Festival, International Journal of Computer Applications, 86(8), 2014, 30-35.
[7] Sridhar Krishna, N., Hema A. Murthy, Timothy A. Gonsalves, "Text-to-Speech in Indian Languages.," International Conference on Natural Language Processing, Mumbai, India, 2002, 317-326
[8] Alam Firoj, S.M.M. Habib, Khan Mumit, Bangla text to speech using festival, Conference on Human Language Technology for Development, Alexandria, Egypt, 2011, 154-161.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Pansy Nandwani. “Speech Synthesis for Punjabi Language Using Festival.” IOSR Journal of Computer Engineering (IOSR-JCE), vol. 19, no. 5, 2017, pp. 14–21.