Text To speech (TTS) Indian Languages: A survey

TEXT TO SPEECH (TTS) IN INDIAN LANGUAGES: A SURVEY

Ranjeet Singh, Santosh Kumar Upadhyay
Ranjeet.sobi@gmail.com, ersk2006@gmail.com
M.Tech Scholar (CSE) Mewar University Chittogarh (Rajasthan), Assistant Prof. Galgotia College of Engineering & Technology Greater Noida

Abstract— Text to speech in Indian language survey paper is, a research overview of work done by the researchers, scholars and Engineers in the field of the processing and building of Speech synthesis. This paper gives an idea of the different approaches, methodologies and techniques taken by them to convert Indian language text data into speech with their respective tools. The quality and methodology of TTS systems can be effectively assessed and test on the basis of reliable, verified and valid listening tests. These works done by various researchers have contributed highly in the technical world and helped building the knowledge of the common people; mainly for visually impaired people. In this paper a detailed overview survey of challenge and technology used by them for Indian language Text to Speech systems is given.

1. INTRODUCTION

A Text to Speech is a computer based speech Synthesis System, abbreviated as TTS, is a form of speech synthesis that converts digital input text into spoken voice output.

The Text to Speech System can be used for various applications. TTS Software embedded with Screen Reader help the visually challenged people to read the text on the computer screen and they would be able to perform the computer operations. Another crucial use is in reading any online service / application for the blind, where a system would process text from an online source and convert it into a speech format. Nowadays, quite sophisticated and high end systems exist that facilitate human–computer interaction for the blind, in which the TTS can help the user, navigate around a desktop system. Apart from this, TTS have been used for weather forecasting, navigation, education and other wide variety of applications.[1,2]

2. Players in TTS field

There are few organizations working in the area of Indian language TTS. Some of the major players in this field are:

- Indian Languages (Hindi, Bengali, Marathi, Tamil, Telugu and Malayalam) Screen Reader application Developed by the following organizations in consortium mode : IIT Madras , IIIT Hyderabad, IIT Kharagpur, CDAC Mumbai, CDAC Thiruvananthapuram
- JNU University Language (Sanskrit)
- Media Lab Asia Shruti-Drishti (Text-to-Speech and Text-to-Braille)
- HP Lab –Hindi TTS based on Festival framework
- Hyderabad Central University (HCU) -Vaani -Telugu
- IIT Kharagpur - An Indian language TTS synthesizer (named SHRUTI) that accepts text inputs in two Indian languages namely Hindi and Bengali
- Simputer Trust – Dhvani TTS [Dhvani, 2001], [Hariharan, 2002]
- C-DAC Bangalore - Matrubhasha API [Raman, 2004]
- IISC Banglore-Thirukkral & Vaachaka,
- The TTS and screen reader technology may also be deployed in CSCs and E-Governance.

There are some other institutions also, where the research on speech synthesis systems is going on, it includes – HCU, IISc Bangalore, Utkal University, TIFR Mumbai, C-DAC Noida and College of Engineering, Guindy etc. [3,4,5]

3. Devanagari based Text to Speech Softwares:

The following three Text to Speech software are available on http://tdil-dc.in/ ,

3.1 Text to Speech screen reader application, for both Windows (NVDA) and Linux (OCRA) based operating systems available in six IL like Hindi, Marathi, Bengali, Tamil, Telugu and Malayalam. Text to speech as a screen reader application which works with different editors like MS -Word, Notepad, Word Pad but it does not support PDF, Currently this software is monolingual. TTS system is independent of any font. TTS system simply reads out what is written in the text editor and it does not support proof reading. [5]

3.2 "Text to Speech Mozilla Browser Plugin": Text to speech software is available freely in the form of a Mozilla plugin.TTS Mozilla plug-in is available in 8 different Indian languages (Hindi, Marathi, Bengali, Gujarati, Malayalam, Tamil, Telugu, Kannada) and also includes English in 3 different accents (Hindi, Tamil, Telugu). Speech change features also provided, with this listener can adjust the reading speed as per his listening. 5 options of voice speed :normal, slow, slower, fast, faster [6].

3.3 Text to Speech Chrome Browser Plug-in: Text to Speech Chrome Plug-in is similar to Text to Speech Mozilla Plug-in. After successful installation of chrome plug-in an icon will appear on top right corner of the browser. Click on the icon after selecting the text and speech file (mp3) will get played. [7]

3.4 Text to Speech android app: SMS Reader on android platform for 5 Indian Languages namely Hindi, Marathi, Tamil, Telugu and Gujarati have been developed and made available through E-gov mobile seva gateway (http://mgov.gov.in). **[9]**

3.5 JNU Sanskrit Tool [8] : The "Samvacaka - A Speech Synthesis System for Sanskrit Prose" is a result of the research carried out by Diwakar Mishra (Ph.D. 2009-2013) under the supervision of Dr. Girish Nath Jha and Dr. Kalika Bali. The application takes simple Sanskrit text and returns synthesized speech or Sanskrit voice output. It does not take into consideration prosodic features and can also be used for Hindi with reasonable success.

3.6 Issues

a. Currently TTS is in development stage, if compared to human voice the speech output of TTS is quite robotic. There is a need for improvement in the naturalness of the TTS.
b. TTS failed to pronounce the percentage sign, Decimal, Hindi numerals, Roman numbers, double quotes.
c. Abbreviations was not handled, e.g. 10gm,1kg etc.
d. Mathematical equation are not handled.
e. Currency symbol like ₹, $ are being skipped by the systems.

4. Challenges in TTS

4.1 Text Preprocessing And Text normalization Challenges: Text preprocessing is a difficult task for Devanagari Text to Speech. And it is more complex in other IL text to speech.

Following Challenges we face in Text preprocessing for Indian language

a. Digits and numerals: **एक इंच में** 254. cm होते है . Here **[इंच]** is read as [Inch], If **[in]** is there it should also be read as [Inch] and in second verse cm is read as ['सेंटीमीटर']

b. Fraction/Dates are problematic. Examples are

- 23/24 can be read as [तेइस बटे चौबीस]

c. Roman Numerals can be a bit problematic. Examples are:

- एलिजाबेथ IV can be read as [एलिजाबेथ चतुर्थ]
- And IV can be read as [चतुर्थ अध्याय]Prefix and postfix will make it more complicated

d. Abbreviations

- 1^{st} /2^{nd}/3^{rd} as प्र. / द्वि / तृ

4.2 Phonetic Analysis Challenge: Phonetic Analysis converts the orthographical symbols into phonological using Phoneme Conversion. Two types of approaches are used in the pronunciation of the word in speech synthesis

A. Dictionary based: A large dictionary is used to store all kinds of words and its pronunciation. The System looks for the word and its respective pronunciation in the dictionary one by one. This type of approach is very fast and the result came, is of better quality. The system has drawback also, like if the word is not found in the dictionary, the system throws an error and the system will stop working.

B. Rule based: In Rule based approach, there is some rule for the letter sounds for a word, letter sounds are blended together to form a pronunciation based on some rule. The System has advantages over previous approach that it requires no database and it can work on any type of input. The system has complexity for the irregular inputs In Indian Language. Aksharas are used to form a word and hence speech data. The Aksharas has following properties

1. An Akshara is an orthographic representation of speech sound in Indian language
2. Akshara are syllabic in nature
3. The Typical forms are Aksharas are V, CV, CCV, CCCV or we have the general form as C*V [1]

4.3 Prosodic Modeling and Intonation: Prosodic Modeling and Intonation describes the Pitch, Stress Pattern, rhythm and intonation in the output speech. Prosodic modeling describes the human emotions. If the system identifies emotions in the vocal and it is very natural synthesized speech. Capturing emotions, stress, rhythm is no doubt a challenging task of voice syntheses for Indian Languages. And it's vital for smart and naturalness of the speech.

Following are the Challenges in the Prosodic Modeling and Intonation:

In voice recording of voice samples. Suppose the person who is recording is not smiling or stressed as required by the sentence so sample voice may not record the smiling emotion or stress emotion.

Example:

a. भारत माता की जय !! – This recording is recorded in full emotion. If the same is recorded in Smiling emotions, then the effect is completely different

b. आप कमरे से बहार जाए – In this the sample is recorded with angry emotion, if the system doesn't catch and it will not up to the output as desired. The modeling of the intonation is to take

care of the modification of the One of the related issues is modification of the pitch contour of the sentence, depending upon whether it is an affirmative, interrogative or exclamatory sentence.[2]

5. Some basic TTS frameworks

This section gives details of the generic frameworks available for the development of a TTS synthesizer. Some of these are as back-end engines and others are full-featured commercial TTS frameworks.[3]

a. Festival TTS Framework The Festival TTS synthesizer was developed in CSTR at the University of Edinburgh by Alan Black and Paul Taylor and in co-operation with CHATR, Japan [Black et al., 2001]. Festival is multi-lingual (currently English, Spanish and Welsh) and is freely available complete diphone concatenation and unit selection TTS synthesizer. *It is the* freeware synthesis system and it includes a comprehensive manual. Festival offers a general framework for developing speech synthesis systems as well as including examples of different modules. As fully, it offers full TTS synthesizer through a number of APIs. The English version of current Festival version is more advanced and the developments for this version are very fast. *The synthesizer is written in C++ and uses the Edinburgh Speech Tools for low-level architecture and has a Scheme (SIOD) - based command interpreter for control*. The latest details and a full software distribution of the framework are available through the Festival Website [Black et al., 2001]. [3,10,11,12,17]

b. MBROLA SYNTHESIZER MBROLA is a better-quality, diphone-based speech synthesizer and is available in the public domain. It is provided by the TCTS Lab of the Faculte Polytechnique de Mons (Belgium) whose main objective it to obtain a set of speech synthesizers for as many languages as possible. The MBROLA speech synthesizer is free of charge for non-commercial, non-military applications. MBROLA database is prepared using any of the recordings in user's speech. Presently there are diphone databases existing for several languages: American English, Brazilian Portuguese, Breton, British English, Dutch, French, German, Greek, Romanian, Spanish and Swedish.
TCTS also provides speech database labeling software: MBROLIGN, a fast MBROLA-based TTS aligner. MBROLIGN can also be used to produce input files for the MBROLA v2.05 speech synthesizer. Demo, comparison of different voices and languages between MBROLA and other synthesis methods can be found on the MBROLA project home page [MBROLA, 1998]. [3]

c. Flite (Festival-lite) is a smaller, faster alternative version of Festival designed for embedded systems and high volume servers.

More information is available at: http://www.speech.cs.cmu.edu/flite/

6. Tools available for development of a TTS synthesizer

Different API's are available for developing a TTS synthesizer [3] provided by different vendors, and different markup languages. There exist many different APIs for speech output but the Microsoft API for synthesizers running on Windows is getting popularity. Another API that is not so frequently used is the Sun-Java Speech API. These two are described below.

a. The Java Speech API: is being developed to allow Java applications and applets to incorporate speech technology. The API defines a cross-platform API to support command and control recognizers, dictation systems and speech synthesizers. Java Speech Grammar Format provides a cross-platform control of speech recognizers. Java Speech Markup Language provides a cross-platform control of speech synthesizers. Text is provided to a speech synthesizer as a Java

String object. The Java Platform uses the Unicode character set for all strings. Unicode provides excellent multi-lingual support and also includes the full International Phonetic Alphabet (IPA), which can be used to accurately define the pronunciations of each syllable. More information can be found on the Java homepage. http://java.sun.com/products/java-media/speech/.

b. Sapi Microsoft's speech API: The major software and speech building Systems vendors are beginning to support Microsoft's Speech API, or SAPI, which is based on the COM specification and is being adopted as the industry standard. The motive of SAPI is to eventually allow interoperability between the speech engines. The Microsoft Speech API provides applications with the ability to incorporate speech recognition (command & control dictation) or TTS, using either C/C++ or Visual Basic. SAPI follows the OLE Component Object Model (COM) architecture. It is supported by many major speech technology vendors. The major interfaces are:

Voice Commands: high-level speech recognition API for command and control. Voice Text: simple high-level TTS API. The Voice Text object is available in two forms: a standard COM interface IVoiceText and companion interfaces, and also an ActiveX COM object, VtxtAuto.dll Multimedia Audio Objects: audio I/O for microphones, headphones, speakers, telephone lines, files, etc. With the Microsoft Speech SDK, and in particular, the TTS VtxtAuto ActiveX COM object, any developer can create a TTS-enabled application using a few simple commands, such as register and speak. More information can be found on the website.
http://msdn.microsoft.com/library/sdkdoc/sapicom/html/intro2sapi.html

c. Markup Language: The Speech Synthesis Markup Language **(SSML)** specification is the W3C markup language specification that defines XML tags to be used in the speech synthesis system to be used in the different speech parameters. SSML Tag also used to define the information like language, metadata for improving the synthetic speech quality in the voice enabled applications. SSML is a markup language of W3C used to create voice enabled applications with email programs and internet browsers.[3]

6.1 Uses of SSML

i. SSML is also used to develop standalone applications and allow users to use voice commands with various online tasks such as searching the Internet, receiving and responding to emails.

ii. SSML is also used in with Spoken Text Markup Language (STML) and Java Speech Markup Language (JSML).

Example of SSML:

```
<?xml version="1.0"?>
<speak version="1.0"
 xmlns="http://www.w3.org/2001/10/synthesis"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/TR/speech-synthesis/synthesis.xsd" xml:lang="hi-IN">
<lexicon uri="http://www.somelexiconfile.com/lexicon.file"/>
<voice gender="female">
<p> <s>I speak <emphasis>Hindi</emphasis></s> <s>I also speak
<emphasis>Marathi</emphasis></s> </p>
<sub alias="International Phonetic Association">IPA</sub>
</voice>
<audio src="royal.wav"> आप का <emphasis> स्वागत </emphasis> है </audio>
</speak>
```

There are many tags used in the current version 1.1 (2010) of SSML defined under W3C. The two tag extensions to the existing SSML specification in the context of Indian languages are:

Transliterate tag

Text input to be used in the most of the Text to Speech systems are either an English transliteration of the Indian language script or is in Unicode. As there is no standard or uniform transliteration scheme to represent different Indian language. Although there is a popular scheme use is IRANS package. To use this, it is very important to specify whether the input text is Unicode or is transliterated using some transliteration scheme.

Transliterate tag has two attributes

a. Codepage
b. Uri

Example of this tag is

```
<? xml version="1.0"?>
<speak version="1.0" xml:lang="hi-IN">
      <transliterate codepage="1137"> मेरा नाम रणजीत है। </transliterate>

</speak>
```

```
<? xml version="1.0"?>

     <speak version="1.0" xml:lang="en-US">
    <transliterate codepage="1252" uri="http://www.example.com/trans.file"> mera nam ranjeet hai </transliterate>
  </speak>
```

Foreign tag

The current version of SSML Specification has two attributes **(<lexicon>, <phoneme>)** that can be used to find the pronunciation of words or phrases.

a. The **lexicon** tag is used to reference external pronunciation dictionaries that are applicable to entire the document
b. In **phoneme** tag the pronunciation for the word or phrase is specified explicitly.

Therefore, there is a concern to define a tag that can be used to indicate that a certain word or phrase needs to be pronounced using a different pronunciation scheme without having to specify its exact phone sequence.

In this case, the tag would point to a lexicon which is different from the globally specified lexicon for the whole document. Such a tag would be helpful when dealing with foreign language words/phrases embedded in a given language text or even in the case of loan words.

Therefore, there is a tag called **<foreign>**_tag that has two attributes "**lang**" and "**uri**".

Example:

```
<? xml version="1.0"?>
```

<speak version="1.0" xml:lang="en-US">I greeted her with a <foreign lang="in" uri="http://www.example.com/lex.file"> Namaste </foreign> and showed her where she could get a ticket for the movie <foreign lang="in" uri="http://www.example.com/lex.file"> “Jaane bhi do

yaaron”</foreign>

</speak>[13-15]

6.2 Pronunciation Lexicon Specification (PLS)

PLS is created by Voice Browser Working Group of W3C and is a standard of W3C. Current version of PLS 1.0 (2008). The aim of PLS is to design an interoperable specifications of pronunciation information which then can be used for speech technology development. PLS provides the facility of mapping between the words or short phrases, their written representations and pronunciation to be use by speech engines and other applications uses speech engines. XML format is used by PLS and for specific language using the baseline PLS specification of the W3C. This specification provides the possibility of providing multiple pronunciations for the same orthography as well as multiple orthographies against an entry of single pronunciation in the PLS. This will almost cover all homophones and homographs. There is a possibility of incorporating acronyms and abbreviations also by providing them as an alias. PLS specification provides a framework and guideline which can be tailored to the needs of a specific language and consequently the XML tag set can be defined to build the PLS data using IPA as UTF 8 representation. PLS can be used by Text to Speech (TTS) and Automatic Speech Recognition (ASR) Engines and can have a wide variety of applications like voice browsers, pedagogical tools etc.[16]

Multiple Pronunciations for the same Orthography in Hindi: For ASR systems it is common to rely on multiple pronunciations of the same word or phrase in order to cope with variations of pronunciation within a language. In the Pronunciation Lexicon language, multiple pronunciations are represented by more than one <phoneme> (or <alias>) element within the same <lexeme> element. In the following example the word "pran [प्राण]" has two possible pronunciations.

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
alphabet="ipa" xml:lang="en-GB">
<lexeme>
<grapheme> pran </grapheme>
<phoneme> praan </phoneme>
</lexeme>
</lexicon>
```

Homograph

Some languages have words with different meanings but the same spelling (and sometimes different pronunciations), called homographs.

For example, in Hindi the word हार (पराजित होना) and the word हार (गहना) have identical spellings but different meanings. It is recommended that these words be represented using separate <lexeme> elements that are distinguished by different values of the role attribute, if a pronunciation lexicon author does not want to distinguish between the two words they could simply be represented as alternative pronunciations within the same <lexeme> element. In the latter case the TTS processor will not be able to distinguish when to apply the first or the second transcription. In this example the pronunciations of the homograph "हार " are shown.

```
<?xml version="1.0" encoding="UTF-8"?>

<lexicon version="1.0"
xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
alphabet="ipa" xml:lang="hi-IN">
<lexeme>
<grapheme> हार </grapheme>
<phoneme> हार </phoneme>
<phoneme> हार </phoneme>
</lexeme>
</lexicon>
```

Pronunciation by Orthography (Acronyms, Abbreviations, etc.)

For some words and phrases, pronunciation can be expressed quickly and conveniently as a sequence of other orthographies. The developer is not required to have linguistic knowledge, but instead makes use of the pronunciations that are already expected to be available. To express pronunciations using other orthographies the <alias> element may be used. This feature may be very useful to deal with an acronym expansion.

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
alphabet="ipa" xml:lang="hi-IN">
<!-- Acronym expansion -->
<lexeme>
<grapheme> रू </grapheme>
<alias>  रुपये </alias>
</lexeme>
<!-- number representation -->
<lexeme>
<grapheme>500</grapheme>
```

<alias> पांच सौ रूपये </alias>
</lexeme>
<!-- Crude pronunciation mechanism and acronym expansion -->
<lexeme>
<grapheme>C DAC</grapheme>
<alias> सी-डैक </alias>
</lexeme>
</lexicon>[13-15]

Conclusion

There are only a few players in the field of IL TTS and most of the product are in beta (under development) version. In this survey we found that there are many issues that need to be addressed to improve the quality and usability of Indian Languages TTS from user point of view. At present TTS voice is quite robotic in comparison to the human voice. There is a need for improvement in the accuracy & naturalness of TTS.

Acknowledgment

I would like to thank all my seniors and colleagues for their direct or indirect help and inspiration, without which this work cannot be completed.

References

[1] Taylor, Paul. Text-to-speech synthesis. Cambridge university press, 2009.

[2] Speech synthesis [http://en.wikipedia.org/wiki/Speech_synthesis]

[3] Gera, Pardeep. "Text to speech synthesis for Punjabi language." *Tech Thesis, Thapar University* (2006).

[4] "Hindi Text To Speech System" [http://blissit.org/hinditexttospeech.htm]

[5] "Indian Languages Technology Proliferation & Deployment Centre" [http://tdil-dc.in]

[6] "Text to Speech Mozilla Browser Plugin" [http://tdil-dc.in/index.php?option=com_download&task=showresourceDetails&toolid=1002&lang=en]

[7] "Text to Speech Chrome Browser Plugin" [http://tdil-dc.in/index.php?option=com_download&task=showresourceDetails&toolid=1551&lang=en]

[8] "Sanskrit Text to Speech (TTS) System- संस्कृत संवाचक "

[http://sanskrit.jnu.ac.in/samvacaka/index.jsp]

[9] "Sandesh Pathak" [https://apps.mgov.gov.in/descp.do?appid=527&action=0]

[10] [Black et al., 2001] Black A, Taylor P, Caley R (2001) The Festival speech synthesis system: system documentation. University of Edinburgh

[11] [Black et al., 2001] User Manual for the Festival Speech Synthesis System,version1.4.3http://fife.speech.cs.cmu.edu/festival/cstr/festival/1.4.3/

[12] " Text - To - Speech Synthesis For Indian Languages & Indian English" [http://www.iitm.ac.in/donlab/website_files/research/Speech/TTS/contents/main.html]

[13] "Pronunciation Lexicon Specification- W3C Working Draft" [http://tdil.mit.gov.in/wsi/docs/PLexicon1.pdf]

[14] "Using SSML for Indian Languages Text to Speech Synthesis Position paper for SSML workshop" [http://www.w3.org/2006/10/SSML/papers/SSML_Paper.pdf]

[15] "Speech Synthesis Markup Language (SSML) requirements" [http://www.tdil.mit.gov.in/WSI/docs/SSML%20doc%201.pdf]

[16] Swaran, Lata. "Challenges for Design of Pronunciation Lexicon Specification (PLS) for Punjabi Language." (2011).

[17] Thomas, Samuel. "Natural sounding text-to-speech synthesis based on syllable-like units." Master's thesis, IIT Madras (2007).