A STUDY OF TEXT TO SPEECH SYSTEMS FOR NON-ENGLISH LANGUAGES

[1]Kurian Benoy, [2]Jiby J Puthiyidam
[1 2] Govt. Model Engineering College, Ernakulam, Kerala

Abstract: Text to speech systems convert any written text into spoken speech. Text-to-speech systems is a vital step for accessibility to disabled people like blind and deaf. It can be used in educational applications as well. Most of the text-to-speech systems are currently made for English language. The objective of this paper is to provide an overview of existing Text-To-Speech synthesis techniques and to provide a comprehensive survey on text-to-speech systems for non-English languages having lesser resources.

Index Terms **– Text-to-Speech, Synthesis techniques, Deep Learning**

I. INTRODUCTION

A text-to-speech (TTS) system [1] converts normal language text into speech. The ultimate goal is to create a natural sound from normal text. The current trend in TTS research calls for systems that enable production of speech in different styles with different speaker characteristics and even emotions. Speech synthesis generally refers to the artificial generation of human voice - either in the form of speech or in other forms like songs etc. The computer system used for speech synthesis is known as speech synthesizer.
A text-to-speech-system is composed of two parts:

- The front-end consists of converting a text into graphemes after text normalization, pre-processing, or tokenisation
- The back-end, referred to as the synthesizer, which converts the symbolic linguistic representation into sound. We can add pitch, prosody etc. to target speech as required.

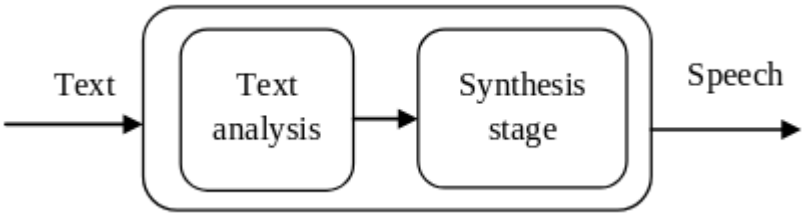

Figure 1 - Phases of Text to Speech System [21]

First, we mention some of the historical developments of Speech synthesis which paved the way for development of contemporary text-to-speech systems. The following contents are extracted from [1] which did a comprehensive historical study of how present system's work.

In 1779, the German-Danish scientist Christian Gottlieb Kratzenstein received the first prize in a competition declared by the Russian Imperial Academy of Sciences and Arts for the models he had designed for the human vocal tract that could generate the five long vowel sounds (International Phonetic Alphabet Notation: [a], [e], [i], [o] and [u]). The bellows-operated "acoustic-mechanical speech machine" by Wolfgang von Kempelen of Pressburg, Hungary, described in a 1791 article[2], followed by adding models of tongues and lips. This allowed it to produce consonants as well as voices. Charles Wheatstone created a "talking machine" based on von Kempelen's design in 1837. Wheatstone's model was a bit more complicated and was capable of producing vowels and most of the consonant sounds. Some sound combinations and even full words were also possible to produce. Vowels were produced with vibrating reed and all passages were closed. Resonances were affected by the leather resonator like in von Kempelen's machine. Consonants, including nasals, were produced with turbulent flow through a suitable passage with reed-off. Joseph Faber exhibited the "Euphonia" in 1846. Paget revived Wheatstone's concept in 1923.

In the 1930s, Bell Labs developed a vocoder that automatically analyzed speech in its fundamental tones and resonances. Homer Dudley developed a keyboard-operated voice-synthesizer called The Voder (Voice Demonstrator), which he exhibited at the 1939 New York World Fair. Dr. Franklin S. Cooper and his colleagues at the Haskins Laboratories designed the Pattern Playback in the late 1940s and completed it in 1950. There have been several different versions of this hardware device; only one currently survives. It reconverted recorded spectrogram patterns into sounds, either in original or modified form. The spectrogram patterns were recorded optically on the transparent belt

The first formant synthesizer, PAT (Parametric Artificial Talker), was introduced by Walter Lawrence in 1953 (Klatt 1987). PAT consisted of three electronic formant resonators connected in parallel. The input signal was either a buzz or noise. A moving glass slide was used to convert painted patterns into six time functions to control the three formant frequencies, voicing amplitude, fundamental frequency, and noise amplitude (track 03). At about the same time Gunnar Fant introduced the first cascade formant synthesizer OVE I (Orator Verbis Electris) which consisted of formant resonators connected in cascade (track 04). Ten years later, in 1962, Fant and Martony introduced an improved OVE II synthesizer, which consisted of separate parts to model the transfer function of the vocal tract for vowels, nasals, and obstruent consonants. Possible excitations were voicing, aspiration noise, and frication noise. The OVE projects were followed by OVE III and GLOVE at the Kungliga Tekniska HÃ¶gskolan (KTH), Swede. (as mentioned in [1])

For a good TTS it's necessary to have text normalization. Text normalization is the process of converting non-standard words (NSWs) such as numbers, and abbreviations into standard words so that their pronunciations can be derived by a typical means (usually lexicon lookups). Text normalization is, thus, an important component of any text-to-speech (TTS) system. Without text normalization, the resulting voice may sound unintelligent.

This paper is structured as following. First, we talk about the prominent methods of Text to Speech in this era and this section captures most of the recent advances in TTS systems which came due to the advent of Deep Learning. This will help any practitioners to understand the various techniques in this field. As our research interests lies in TTS systems in Non-English languages, we have focused on selecting

the most appropriate methods of TTS for such languages. We have mentioned some of the prominent TTS methods implemented in low resource Non-English languages in the related works section. Finally, we talk about the experimental results of selected TTS papers and our future plans.

II. METHODS OF TEXT-TO-SPEECH

In this section we discuss some of the prominent methods of Text-to-Speech generation in the contemporary world. We have included some of the most popular and innovative techniques which came in the near future as well here.

A. Concatenative Synthesis

Synthesis of unit selection allows use of broad recorded speech repositories. Each documented utterance is segmented into some or all of the following during the compilation of a database: individual phones, diphones, half-phones, syllables, morphemes, verbs, phrases, and sentences. Typically, division into segments is done using a specially modified speech detector set to a forced alignment mode with some manual configuration. So concatenative speech synthesis method involves the production of artificial human like speech from pre-recorded units of speech by phonemes, diphones, syllabus, words or sentences.[6]

B. Diphone synthesis

Diphone synthesis uses a minimal speech database containing all the diphones (sound-to-sound transitions/combinations) that occur in a language. The number of diphones depends on the phonotactics of the language: for example, there are about 800 diphones in Spanish and about 2500 in German. As only single instances of speech are available in speech database, in order to obtain good quality of synthesised speech with prosody and naturalness, speech processing techniques are applied. PSOLA, TD-PSOLA, LP-PSOLA, ESNOLA, FD-SOLA are some of the common techniques used for obtaining good quality synthesised speech.[6]

C. Formant synthesis

Formant Synthesis does not use human speech samples during run time. Instead, a synthesized speech output is generated using an additive synthesis and an acoustic model (physical modeling synthesis). Parameters such as fundamental frequency, voice and noise levels vary over time are used to create a waveform of artificial speech. This method is sometimes referred to as a rule-based synthesis; however, many concatenative systems also have a rule-based component. Many systems based on formative synthesis technology generate artificial, robotic-sounding speech that would never be mistaken for human speech. Formative synthesis systems have advantages over concatenative systems. Formant synthesized speech can be accurately intelligible, even at very high speeds, eliminating acoustic errors that often affect concatenative systems. High-speed, visually impaired speech is used to easily access devices using a screen reader. Formant synthesizers are usually smaller than Concatenative speech synthesis. [7]

D. HMM based synthesis

Hidden Markov Model (HMM) based synthesis is a synthesis process based on secret Markov models, and is also known as the statistical parametric synthesis. In this method, the frequency spectrum (vocal tract), the fundamental frequency (voice source), the length of the source (prosody) of speech are simultaneously modeled by HMMs. Speech waveforms are generated from HMMs themselves on the basis of the maximum probability criterion.[12]

E. Articulatory synthesis

Articulatory synthesis refers to computational techniques for speech synthesis based on models of the human vocal tract and the articulation processes that occur there. The first articulatory synthesizer to be used routinely for laboratory experiments was created at Haskins Laboratories in the mid-1970s by Philip Rubin, Tom Baer and Paul Mermelstein. This synthesizer, known as ASY, was based on the vocal tract models created by Paul Mermelstein, Cecil Coker, and colleagues at Bell Laboratories in the 1960s and 1970s. Articulatory speech synthesis refers to make speech based on models of human vocal tract and articulation process.[8]

Articulation synthesis comprises of:

1) Generation of vocal tract movements
2) Convert movement information into continuous succession of vocal tract geometries
3) Generate acoustic signals from articulatory information

F. Hybrid Text to Speech synthesis

The Hybrid TTS approach is a combination of the two main approaches of synthesis: Concatenative synthesis and Statistical Synthesis. The hybrid TTS combines the characteristics of smooth transitions between adjacent speech segments of a Statistical TTS with the naturalness of a Concatenative TTS. This is achieved by interweaving natural speech segments and statistically generated speech segments. The statistical segments are positioned so as to smooth discontinuities in the synthesized speech, while enabling as far as possible natural speech sequences as they appear in the training inventory disadvantages of this system are the degradation in speech quality when TTS speech inventory is small and more signal processing requirement [9].

G. Statistical Parametric Synthesis

Statistical parametric synthesis makes use of averaged acoustic inventories that are extracted from the speech corpus.The extracted parameters of speech are the spectral parameters such are cepstral coefficients or line spectral pairs, and excitation parameters such as fundamental frequency. Statistical Parametric synthesis has the advantages of requiring less memory to store the parameters of the model, rather than the data itself and it allows more Variation in the speech produced for example, an original voice can be converted into another voice. The most commonly used statistical parametric speech synthesis technique is the Hidden Markov Model (HMM) synthesis and Unit Selection synthesis [10].

F. Efficient speech

A new model for audio synthesis WaveRNN is introduced in [4]. A single layer RNN with a softmax layer WaveRNN matches current SOTA of Wavenet models. Yet it's possible to reduce the N (no of layers) and operations in GPU. A technique of weight pruning is applied to reduce number of weights in WaveRNN. A sparse WaveRNN makes it possible to sample high fidelity audio on a mobile CPU in real time. A new generation scheme based on sub scaling that folds a long sequence into batches of shorter sequences and allows one to generate multiple samples at a time. The subscale WaveRNN produces 10 samples per step without loss of quality and offers an orthogonal method for increasing sampling efficiency.

WaveRNN demonstrates that a simple recurrent neural network for sequential modelling of high fidelity audio, and demonstrated a high performance implementation of this model on GPUs. It's shown that large sparse models have much better quality than small dense models with the same number of parameters and we have written high performance block-sparse matrix-vector product operations to demonstrate that sampling time is proportional to parameter count. It showed that high fidelity audio generation is now achievable on widely available low-power mobile CPUs. Finally, this paper introduced the subscale dependency scheme that lets sequential models generate many samples per step while preserving the output quality of the original model. The underlying ideas of the methods we introduce are not specific to audio, and the results of sparse models have implications for inference in all types of neural networks.

G. Wavenet Based Models

In September 2016, Deep Mind proposed WaveNet, a deep generative model of raw audio waveforms [3]. This shows the community that deep learning-based models have the capability to model raw waveforms and perform well on generating speech from acoustic features like spectrograms or spectrograms in Mel scale, or even from some preprocessed linguistic features. The model is fully probabilistic and self-regressive, with the predictive distribution of each audio sample conditioned on all previous ones. Data with tens of thousands of samples per second of audio can be efficiently trained. Applied to text-to-speech, it produces state-of - the-art performance, with human listeners rating it as significantly more natural than the best sound parametric and concatenative systems can produce for both English and Mandarin. Wavenet model triggered a huge development of Text to speech architectures based on Deep Learning which is mentioned in [11].

H. Fast Speech

Compared with traditional concatenative and statistical parametric approaches, neural network based end- to-end models suffer from slow inference speed, and the synthesized speech is usually not robust (i.e., some words are skipped or repeated) and lack of controllability (voice speed or prosody control). In this work [5], a novel feed-forward network based on Transformer is used to generate Mel-spectrogram in parallel for TTS. Specifically attention alignments are extracted from an encoder-decoder based teacher model for phoneme duration prediction, which is used by a length regulator to expand the source phoneme sequence to match the length of the targeted Mel-spectrogram sequence for parallel Mel-spectrogram generation. Experiments on the LJ-Speech dataset show that our parallel model matches autoregressive models in terms of speech quality, nearly eliminates the problem of word skipping and repeating in particularly hard cases, and can adjust voice speed smoothly. Most importantly, compared with autoregressive Transformer TTS, our model speeds up Mel-spectrogram generation by 270x and the end-to-end speech synthesis by 38x. Therefore, this method is called Fast speech.

Concatenative Synthesis
Diphone Synthesis
Formant Synthesis
HMM based Synthesis
Articulatory Synthesis
Hybrid Text to Speech Synthesis
Statistical Parametric Synthesis
Efficient Speech
Wavenet Based Models
Fast Speech

Table 1- Popular TTS Techniques

III. RELATED WORKS

A. Text Normalization for Bangla, Khmer, Nepali, Javanese, Sinhala and Sudanese TTS [13]

The main contributions of this [13] paper are as follows:

1. Describe a general method for working with native speakers in identifying patterns and grammars needed to normalize text
2. Making available text normalization grammars and their test cases for a wide range of common semiotic classes for 6 low resourced languages
3. Provide a recipe for utilizing these grammars and for integrating them into actual text-to-speech systems.

Text normalization takes plain text as input for any language. It takes input words being separated each other using whitespaces while for languages which do not use whitespaces to separate works - like Khmer, the input is the output after passing text into word segmenter. Text normalization is divided into two phases. First, input text is analyzed and NSWs are classified into semiotic classes. In this phase, some input tokens may be grouped together. For example, input text “15 km” may be grouped together and are classified as a measurement token. Then, verbalizer grammar for each semiotic class will convert the classified NSWs into standard text accordingly

After identifying the semiotic classes, they reached out to Native speakers in each language with a questionnaire. The questionnaire contains a set of questions for each semiotic class. The questions were designed to capture the writing and verbalizing system of the language. They took into consideration of various language-specific characteristics that affect text normalization. For each language there are particular characteristics for the same. In case of Bangla: different inflection cases are handled and 4 different time indicators similar to “am/pm”. Besides the above considerations, all languages in this set, other than Javanese and Sudanese, have their own alphabets and digits. Bangla uses the Bengali alphabet. Khmer uses the Khmer script. Nepali uses Devanagari and Sinhala uses the Sinhala alphabet. Khmer writing contains inconsistent usages of the zero-width space character. Bengali, Nepali and Sinhala also utilize zero-width non-

joiner (U+200C). Various unit test cases was created for the grammars so as to solicit both common and corner cases. So our test cases verify if everything works fine.

On obtaining test cases and explanations about how to classify and verbalize different semiotic classes, create grammar rules to do the classification and verbalization. The grammars that created is Thrax grammars, which consist of mostly regular expressions and context-dependent rewrite rules. The grammars can then be compiled with the Thrax grammar compiler, which turns them into archives of finite state transducers. First, the input text will be classified into different semiotic classes. The output of this classification step is a protocol buffer which contains information about the original token and its semiotic class. Then, the protocol buffer will be passed to the verbalizer component, which converts the token into one or more standard words based on the verbalization rules for the semiotic class associated with the token. For example, the classification of input"100 m" will output the following protocol buffer: measure decimal part:"100" units:"meter"

The usage of text normalization with open source tools by Google is mentioned properly in this paper [13]. Moreover tools are made open source and presented text normalization grammars, both classifiers and verbalizers, for six low resource languages and along with test cases are available free to public. The process of testing and utilizing this grammar is mentioned in Sparrow hawk text normalization system. TTS build using this techniques can be widely adopted for low-resource languages and process is universally applicable in any language. These text normalization techniques helped Google in successfully creating the Project Unison, the first text to speech system in Bangla [19]. More details on the experiences have been mentioned in detail in [20] and about how to tackle the issue of solving Text-to-speech systems for low resource languages.

B. Corpus Driven Malayalam Text-to-Speech Synthesis for Interactive Voice Response System [14]

In paper [14] written by Arun Soman, etc., a corpus-driven Malayalam text-to-speech (TTS) system based on the concatenative synthesis approach is explained. The most important qualities of a synthesized speech are naturalness and intelligibility. In this system, words and syllables are used as the basic units for synthesis. The corpus consists of speech waveforms that are collected for most frequently used words in different domains. The speaker is selected through subjective and objective evaluation of natural and synthesized waveform. The proposed Malayalam text-to-speech system is implemented in Java multimedia framework (JMF) and runs on both in Windows and in Linux platforms. The proposed system provides utility to save the synthesized output. The output generated by the proposed Malayalam text-to-speech synthesis system resembles natural human voice. Text to speech reader software converts a Malayalam text to speech wav file that has high rates of intelligibility and comprehensibility.

The corpus was collected by a single best speaker wave form. The best speaker means the speech produced by that speaker have capabilities with respect to energy profile, speaking rate, pronunciation and into nation. Input Malayalam text was first text normalized which made the input text words in non- standard form like Numbers, dates, etc. and punctuations were also removed. Then process of sentence splitting took place in the paragraphs and words were separated out. Romanization is the representation of written word with a roman alphabet. In this system Romanized form of Malayalam words/syllables are generated. For representing the written text the method used for Romanization is transliteration and for spoken word, the method is transcription. The final stage is the concatenation process.

All the arranged speech units are concatenated using a concatenation algorithm. The concatenation of speech files is done in java media framework. The main problem in concatenation process is that there will be glitches in the joint. The concatenation process combine all the speech file which is given as a output of the unit selection process and then making in to a single speech file. This can be played and stopped any where needed.

C. Orthography with Maratha [15]

Speech synthesis models are typically built from a corpus of speech that has accurate transcriptions. Paper [15] consider many of the languages of the world do not have a standardized writing system. This paper is an initial attempt at building synthetic voices for such languages. It may seem useless to develop a text-to-speech system when there is no text available. A novel method to build synthetic voices from only speech data is shown here. Experimental results and oracle studies shows that it is possible to automatically devise an artificial writing system for these languages, and build synthetic voices that are understandable and usable.

The speech data in a target language with no well-defined orthography for transcriptions is given. A simple method to deal with this situation is to run an automatic speech recognizer over available speech data and use its output as transcriptions. The caveat with using a speech recognizer is that because our target language does not have a text form, a speech recognizer will not exist in that language. We hence have to use a speech recognizer in another language: a language that has an orthography, and large corpora to train speech recognizers. This presents another caveat: that is recognizing in a different language than the models are trained for. Using the default language model is thus not ideal, and we need to adapt it so that it is suitable for our target language. It also uses phonetic decoding instead of word level decoding.

The paper [15] solution proposes the following:

1) Choose an appropriate acoustic model for speech recognition
2) Choose a language that has an orthography and is phonetically close to our target language, and then build a phonetic language model on text in the language.
3) Run phonetic decoder on our target speech data with these two models and obtain transcripts
4) Build voice using the speech data and the phonetic transcripts just obtained.

The paper has addressed a novel problem of building speech synthesizers for languages without an orthography. In the solution proposed automatically developing a writing system for the language, using a speech recognition system. The iterative method to build targeted acoustic models yield very good improvements in synthesis quality. The objective and subjective results, as well as oracle results on Marathi, which show that direction to building synthesis models without written text is promising. We also showed similar results on Hindi and Telugu, thus showing that our methods are language Independent.

D. Maithili Text to Speech system [16]

This paper describes the method of creating a text-to-speech system for Maithili, which is a similar variant of Hindi. As most Indian languages, Maithili is syllabic in nature and concatenative method is used for purpose of speech generation taking speech syllable as the basic unit. The concatenate Unit Selection Synthesis (USS) technique is used. Naturalness of USS for small amount of data is better compared to other methods.

The workflow of this TTS made by Amit Kumar Jha, etc is:

1) Input Maithili text written in Devanigiri script using UTF-16
2) Inputted text is normalized with the help of three
3) Inputted text is segmented into sentence level. Afterwards, it is segmented into word level using white space.
4) A word level search is done in database and if it is found then corresponding speech file is added into playlist. Else, the word is broken into corresponding syllables and corresponding syllables files are searched and added in playlist.
5) Found speech units are concatenated in playlist using digital signal processing.
6) Add prosodic features to the speak file according to the types of sentence.
7) Play the sound of playlist.

The process of text normalization for Maithili language is mentioned in detail in [16]. To increase the naturalness of TTS system, 1055 most frequently occurring word have been recorded and stored in separate lexicon. The system support UTF-16 for text input and a C#.NET interface is used for developing TTS system in Maithili. The speech database consists of 930 syllable (C*V) in total. Each position has 300 syllables and 10 independent vowels. Each position has 300 syllables and 10 independent vowels. 930 units of speech data is build all three positions. This is the first TTS system which exists for Maithili language till date

E. Speaking Style Adaption in Text-to-Speech Synthesis using sequence-to-sequence models with attention [17]

Neural networks which are data driven is good in Text to Speech synthesis. The paper [17] is aimed for challenging speaking styles like Lombard speech (speaking in loud voice) where it's difficult to generate large corpora. A new transfer learning method which adopt Lombard style from Normal speaking style. They use a learning method to adopt sequence to sequence based TTS system of normal speaking style to Lombard style. Moreover on evaluation results indicated that an adaption system with Wavenet Vocoder clearly outperformed conventional deep neural networks based on TTS.

Here TTS system uses a sequence to sequence model with attention. The model accepts either mono-phonemes or graphemes as inputs and emits acoustic parameters as outputs. It consists of three main components: 1) Encoder 2) attention and 3) decoder. The encoder takes text sequence x of length L as input, which represented either in the character or phoneme domain as one-hot vectors. The encoder learns a continuous sequential representation h using various neural network architectures such as LSTMs or CNN. At each output time step t, both the attention and decoder modules work together. The decoder takes the previous hidden state and current context vector as inputs and generates the current output. The process runs until the end of the utterance is reached. The accuracy of recognizing Lombard speech by our TTS is 95%.

F. Indonesian TTS using Diphone based speech TTS [18]

Paper [18] shows a novel approach to create a text to speech system for Indonesian language. This approach first creates a database of Indonesian diphone at first. Indonesian diphone synthesis uses speech segment of recorded voice from text to speech and save it as audio file like WAV, MP3. First a diphone databases is created and then convert text to speech from input of numbers, words. They used diphone concatenative synthesis in which recorded segments were collected.

The two main contributions of this paper are: First developed a diphone database including creating a list of words consisting of diphones organized by prioritizing diphones in this system. Second develop system using Microsoft visual Delphi 6.0, includes: the conversion system from the input of numbers, acronyms, words, and sentences into representations diphone. There are two kinds of conversion (process) alleged in analyzing the Indonesian text-to-speech system. One is to convert the text to be sounded to phoneme and the other convert the phoneme to speech. Method used in this research is called Diphone Concatenative synthesis, in which recorded sound segments are collected. Every segment consists of a diphone (2 phonemes). This synthesizer may produce voice with high level of naturalness. The Indonesian Text to Speech system can differentiate special phonemes like in 'Beda' and 'Bedak' but sample of other specific words is necessary to put into the system. This Indonesia TTS system can handle texts with abbreviation, there is the facility to add such words. Indonesian Text-To-Speech System using diphone concatenative synthesis can produce speech or language the natural approach. Speech or resulted sound may not be perfect for several causes such as poor recording and distorted phoneme segmentation process where front, middle and end borders are not synchronized.

Research paper	Language	Method	Metric	Disadvantages
Corpus Driven Malayalam Text-to-Speech Synthesis for Interactive Voice Response System	Malayalam	Concatenative synthesis	Mean Opinion Score	MOS score for complex sentences are less
Text Normalization for Bangla, Khmer, Nepali, Javanese, Sinhala and Sudanese TTS	Bangla, Khmer, Nepali, Javanese, Sinhala, Sudanese	NA	No. of sentences correct in tests generated by grammar	Need to add coverage of grammar to have more measurement units in each language and solve test abbreviation issues in each language Not able to evaluate these grammar against some standard unseen text corpora
Orthography with Maratha	Maratha, Telegu	Novel technique specific for this paper only	MCD sore	Detecting noise in ASR transcript and mitigating the effects of that noise in synthesis output. Need to have a larger acoustic model trained on larger phone set
Maithili Text to Speech system	Maithili	Unit Selection Synthesis	Mean Opinion Score (82% accuracy)	Haven't added features like prosody and can be made more robust
Speaking Style Adaption in Text-to-Speech Synthesis using sequence-to-sequence models with attention	Lomard speech(speaking in loud noise)	Wavenet model using seq-to-seq RNN	Mean Opinion Score	Not yet trained Wavenet and Seq2Seq TTS model in a single pipeline.
Indonesian TTS using Diphone based speech TTS	Indonesian	Diphone Concatenative synthesis	Mean Opinion Score	The process of finding an example of the word in phoneme is less precise so result is not perfect In segmentation process diphone still much less precise

Table 2 - Comparative Analysis of Related Works

IV. CONCLUSION

On studying the various papers we realized that the process of synthesizing speech from text has evolved rapidly over the years. Studying TTS on various non-English languages it is realized the accuracy and naturalness of TTS systems in these languages are not as good as English TTS. After a detailed study of TTS on various low resourced non-English languages, we have concluded the following:

The paper [14] which worked on concatenative synthesis was used for one of the earliest text to speech system for Malayalam. The experiments in the paper give good MOS score for small and easy sentences (MOS score of 4.0) while for harder sentences the accuracy is giving an MOS score of 2.72. In [15] Text-to speech for languages like Martha based on techniques mentioned in paper[3] on evaluation using MCD score gives considerably good result. The tests were conducted by taking speech data of Telugu and Hindi as well with the assumption that both languages have no orthography and no transitions were provided to them and the accuracy is considerably good. The iterative models mentioned in [15] to build targeted models yield very good improvements in synthesis quality. In case of Maithili Text-to-speech [16], evaluation of TTS depends on three key approaches: 1. User Testing 2. Feature comparisons, and 3.objective measures. The basic approach to measure the speech output was decided by talking people from ten different backgrounds, and calculating Mean opinion score. The output score in average for all the speakers is satisfactorily close to 84% for speech generated by [16]. The intermixing of Hindi and English strings in Maithili language helped to give a better result for this system. In [17] compared different TTS models and vocoders to adapt the speaking style of speech synthesis from normal to Lombard. The study proposes using an adaptation method based on fine-tuning combined with sequence-to-sequence based TTS models and the WaveNet vocoder conditioned using Mel spectrograms. Listening tests show that the proposed method outperformed the previous best method that was developed using a LSTM-RNN based adapted systems. According to results mentioned in paper by Sultarman etc. [18] Indonesian Text- To-Speech System using diphone concatenative synthesis can produce speech or language the natural approach which can differentiate special phonemes like Beda and Bedak and can handle text with abbreviations as well. [13] Which provided a new text normalization system helps in creating better TTS systems for low resource languages in the future.

Based on this survey, we aim to create an accurate and steady Text-to Speech system for Malayalam as a future work. Most of the TTS systems in Malayalam are not so good now as it isn't able to produce speech with decent quality and add prosody features to the sound. There has been no development of TTS in Malayalam for the past 10 years, we hope to make some positive new change in this direction.

REFERENCES

[1] History and Development of Speech Synthesis, Helsinki University of Technology (http://research.spa.aalto.fi/publications/theses/lemmetty_mst/chap2.html)

[2] Mechanismus der menschlichen Sprache nebst der Beschreibung seiner sprechenden Maschine ("Mechanism of the human speech with description of its speaking machine", J. B. Degen, Wien). (in German)

[3] Oord, Aaron & Dieleman, Sander & Zen, Heiga & Simonyan, Karen & Vinyals, Oriol & Graves, Alex & Kalchbrenner, Nal & Senior, Andrew & Kavukcuoglu, Koray. (2016). Wavenet: A generative model for Raw Audio (https://arxiv.org/abs/1609.03499)

[4] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, etc. "Efficient Neural Audio Synthesis", 25 June 2018

[5] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, Tie-Yan Liu. "FastSpeech: Fast, Robust and Controllable Text to Speech." 29 May, 2019

[6] Rubeena A. Khan , J. S. Chitode, "Concatenative Speech Synthesis: A Review, International Journal of Computer Applications" (0975 – 8887). Volume 136 – No.3, February 2016.pg-1 to 4.

[7] Sami Lemmetty. "Review of Speech Synthesis Technology". Helsinki University of Technology, Department of Electrical and Communications Engineering. March 30, 1999.

[8] Pertti Palo. "A Review of Articulatory Speech Synthesis". Espoo, June 5, 2006

[9] Stas Tiomkin, David Malah, Slava Shechtman, and Zvi Kons, "A hybrid text-to-speech system that combines concatenative and statistical synthesis units" IEEE Transactions on Audio, SPEECH, and Language Processing, vol. 19, no. 5, JULY 2011 pp 1278-1288.

[10]Heiga Zen, Keiichi Tokuda, Alan W. Black ,"Statistical parametric speech synthesis", Speech Communication vol.51,no.11,2009,pp. 1039–1064.

[11]http://www.erogol.com/text-speech-deep-learning-architectures/

[12]Raitio, Tuomo, et al. "HMM-based speech synthesis utilizing glottal inverse filtering." Audio, Speech, and Language Processing, IEEE Transactions on vol.19, no.1, 2011, pp. 153-165.

[13]Keshan Sodimana, Pasindu De Silva , Richard Sproat, Theeraphol Wattanavekin , Chenfang Li,Alexander Gutkin, Supheakmungkol Sarin, Knot Pipatsrisawat, "Text Normalization for Bangla, Khmer, Nepali, Javanese and Sudanese, Text-to-speech systems". The 6th Intl. Workshop on Spoken Language Technologies for Under-resourced Languages

[14]Arun Soman, Sachin Kumar S., Hemanth V. K., M. Sabarimalai Manikandan, K. P.Soman. "Corpus Driven Malayalam Text-to-Speech Synthesis for Interactive Voice Response System". International Journal of Computer Applications (0975 – 8887)Volume 29– No.4, September 2011.

[15]Sukhada Palkar, Alan W Black, Alok Parlikar."Text-To-Speech for Languages without an Orthography"

[16]Maithili Text-to-Speech System, Amit Kumar Jha, Piyush Pratap Singh, Pankaj Dwivedi

[17]Bajibabu Bollepalli, Lauri Juvella, Paavo Alku. "SPEAKING STYLE ADAPTATION IN TEXT-TO-SPEECH SYNTHESIS USING SEQUENCE-TO-SEQUENCE MODELS WITH ATTENTION"

[18]Sultarman. "Indonesian Text-to-speech system using Diphone concatenative synthesis"

[19]"Text-to-Speech for Low Resource Languages: But can it say Google? "(https://ai.googleblog.com/2016/02/text-to-speech-for-low-resource.html)

[20]One Down, 299 to go (https://ai.googleblog.com/2018/09/text-to-speech-for-low-resource.html)

[21]Dessai Sidhi, etc Survey on Various Text to speech synthesis