Analysis and implementation of Deep Neural Network based Text to Speech (TTS) systems

Apoorva Mahajan
E&TC Department
Vishwakarma Institute of Technology
Pune, India

Shreya Killedar
E&TC Department
Vishwakarma Institute of Technology
Pune, India

Shoumik Nandi
E&TC Department
Vishwakarma Institute of Technology
Pune, India

Prathamesh Doundkar
E&TC Department
Vishwakarma Institute of Technology
Pune, India

Prof. Dr. Ashutosh Marathe
E&TC Department
Vishwakarma Institute of Technology
Pune, India

***Abstract* –Speech synthesis is the process of converting a written message in form of text to equivalent message in spoken form. A Text-To-Speech (TTS) synthesizer works as a computer-based system that should be able to read text. We propose a text to speech (TTS) method (voiceloop by facebook) that is able to convert text to speech in voices that are sampled in the wild. Unlike other systems, this approach was able to handle unconstrained voice samples and without requiring aligned phonemes or complex linguistic features extractions.It used an attention model to extract the context and also ensured support to multiple speakers. It is a memory efficient solution as a single buffer is used for all the computations.GMM based attention was used for sequence to sequence modelling . Instead of conventional RNNs,a memory buffer was employed in this system. Input encoding part using a context-free lookup table mechanism is an extremely simple approach which was incorporated hence making the model less complex. Finally WORLD vocoder was used for the actual synthesis of the speech from the feature vector.**

Keywords—Attention model, GMM, Vocoder, multi-speaker, buffer

I. INTRODUCTION

Text processing and speech generation are two primary components of a text to speech conversion. The main aim of the text processing component is to operate on the given input text and output appropriate sequences of phonemic units. These phonemic units are converted into the actual speech by the speech generation component either by using parameters or by selection of the best unit from an available speech corpus.Audio reading devices for blind people make use of this technique. In recent times use of text-to-speech conversion technology has gone far beyond the dissabled community to become a major additive to the fast expanding use of digital voice storage for voice mail and voice response systems.

For natural sounding speech synthesis, it is necessary that the text processing component produces a suitable sequence of phonemic units corresponding to input text. With conventional speech synthesis methods, this objective has been achieved by dividing the problem into two stages. The frontend,which converts the text into linguistic features. These linguistic features are phones, syllables, words, phrases and also utterance-level features . Backend (second part), takes linguistic features as input made available by the frontend and produces the corresponding sound.[2] Modern TTS systems are based on complex,numerous stage dependent processing pipelines, each of which may again depend on other custom features and heuristics. This paper presents the modern approach used by facebook voiceloop without requiring aligned phonemes or linguistic features.A more simplified framework using sequence-to-sequence models with attention was proposed for this TTS.

II. LITERATURE REVIEW

Text to speech (TTS) methods are of four types:Articulatory synthesis,formant(rule-based), concatenative, statistical-parametric (mostly HMM based), and neural.

Computational techniques for articulatory synthesis speech based on models of the human vocal tract and the articulation processes occurring there. It is one of the most complex approaches to use and the computational load is also more than with other familiar techniques. Advantages of articulatory synthesis are that the vocal tract models allow accurate modeling of transients due to abrupt area changes.[4].Homer Duley vocoder(1939) and Gunnar Fant's OVE synthesizer(1950) were the models that used this synthesis technique

Formant synthesis starts with acoustics, creates rules/filters to form each formant. It is derived from the concept of source-filter-model of speech. There are two structures namely, parallel and series, but for good performance a combination of these is commonly used. It is not a dynamic method because of the static rules.

Concatenative synthesis uses databases of stored speech to assemble new utterances. Different phonemes concatenated as per text. Diphone and unit selection methods are the broad categories of concatenative synthesis. Diemo Schwarz (2000) used Concatenative data based synthesis technique. In this paper they compared the musical and speech synthesis .The tone, naturalness and intelligibility contrasts were found. Manual based systems for segmentation were used. The distance function was not accurately tuned, there was no practical target related timbre space and phase coupling for speech segmentation and synthesis. The author proved that using the technique of data-driven concatenative synthesis primarily based on varying unit selection to musical synthesis application is a well grounded and efficient method.[6]

Kalyan D. Bamane review showed that the author applied a unit selection algorithm in language synthesis(marathi) ,2011. They had only worked on Marathi Language. The author used the different choice of units like words, Di-phone and tri-phones as databases. In this paper they focused deeply on Di-phone and Tri-phone. They comment that old style TTS is not particularly positive. In this paper they generated speech signals from "from scratch". Here also the most important quality of speech synthesis system is naturalness and intelligibility. They show that the result is 95% quality voice.[7]

Statistical machine translation (SMT): It is distinguished by the use of machine learning methods. SMT is a data-driven method which makes use of simultaneously aligned corpora and considers translation as a mathematical reasoning based problem. In that, each sentence in the output language is a translation with probability from the input language. The more the probability, the higher is the accuracy of translation and the other way around. Catalin Ungurean (2011) had addressed the NLP and TTS. TTS automatic syllabification is necessary for lexical stress assignment, prosody generation and letter to phone conversion of the input text. They had implemented a system by using a hybrid strategy, a minimal set of rules, followed by a data driven approach. They had also used a set of phonetic transcription rules with the correctly syllabified words. Moreover, they demonstrated that lexical stress prediction can help the letter to phone process, by solving some additional ambiguities.[5]

These systems learn a fixed set of speaker embeddings and therefore only support synthesis of voices seen during training. In contrast, VoiceLoop [1] proposed an unique architecture mainly based on a fixed size memory buffer which can generate speech from voices in youtube videos.On the contrary to other TTS models, this network is trained on untranscribed speech containing echoes and background noise from a huge number of speakers. The multi speaker support along with the optimized use for single buffer memory marks the efficiency and accuracy of this model.

III. SYSTEM IMPLEMENTATION

In this system input text is transformed to phoneme encoding and then context vector is created with attention mechanism.Each speaker encoding is done alongside using LUT. With context, speaker ID, previous output, and buffer, the new buffer representation is created with a shallow fully connected neural network and inserted into the buffer memory. Then the output is created by using buffer and speaker ID with another fully connected neural network. A novel speaker can be adapted just by fitting it while fixing all other components.

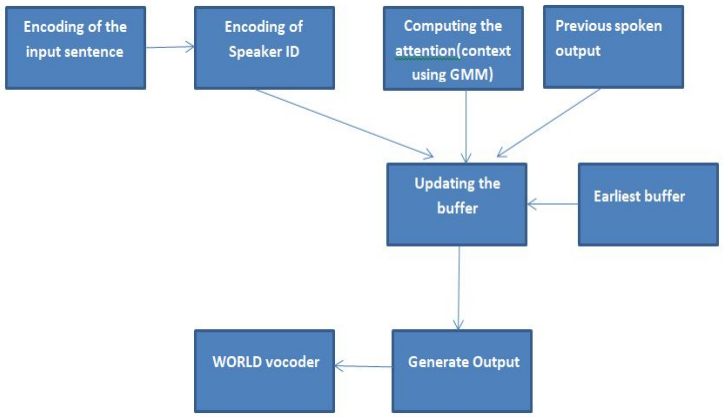

Fig.1. System Implementation of TTS

IV. METHODOLOGY

A.Phonemizer

Phoneme is the smallest unit of sound of a word. To hear these sounds the word is broken down into the sounds using the segmentation process. The phonemizer carries out the process of breaking down the words into phonemes. The input sentence is segmented on the basis of phones, syllables and words using a separator. The text of the sentence along with the separator is passed to the phonemizer. The phonemizer starts processing the

sentence in the phonemize function. The mode allows the user to select between two backends namely Festival and Espeak. The arguments such as the backend to be used, the text to be processed, the language of the input text, the separator and whether parallel processing is to be used or not are passed in phonemize. Three conditions are checked for the further process to be executed: whether or not the backend chosen is Festival or Espeak, whether or not the backend chosen is installed on the system and whether the language of the input text is supported. If any of these conditions are not satisfied an error message is displayed and the program is terminated. If all the conditions are satisfied the backend program is executed. In the model only the festival backend is used since the text is in U.S. english and the festival is more efficient. When festival is the selected backend the segmentation into phonemes takes place in three steps: preprocessing, processing and post-processing. In preprocessing the text is converted into the format needed for the festival script to process it. In processing the text is converted into a SyllStructure tree using the festival script. Finally post-processing is carried out where the phonemes are extracted from the tree. The string of phonemes extracted is converted into a list and passed back. After this the phonemes are mapped with a particular code.

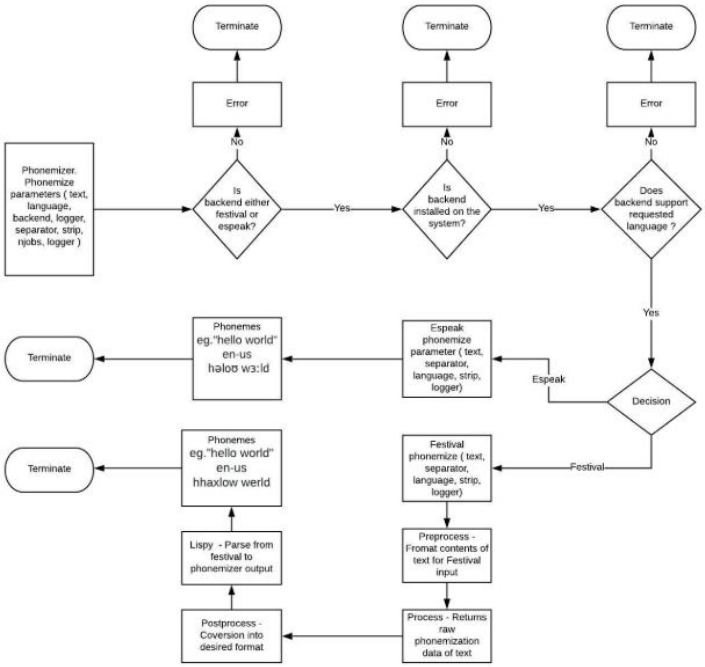

Fig.2. Flow chart for phonemizer

B.Attention Model

The speaker id is mapped (using a lookup table) onto a speaker embedding (Z)which is also trained by the neural network. The attention is computed using Graves Gaussian mixture model(GMM) based monotonic attention mechanism.GMM is a mixture density model ..The basic approach of mixture density networks is to use the outputs of a neural network to parameterise a mixture distribution. The mixture weight outputs are normalised with softmax function to ensure they form a acceptable discrete distribution, and the other outputs are passed through suitable functions to keep their values within meaningful range (for example the exponential function is typically applied to outputs used as scale parameters, which must be positive).Mixture density network are trained by maximising the log probability density of the targets under the induced distributions.[17][18].The prior buffer state is fed to the GMM based attention model which outputs the gaussian parameters.Further the attention weights are calculated by performing operations on the obtained parameters.These weights are multiplied with input embedding to give the context vector as the output .[2][16].At each time step a new frame is generated using Nu, which takes as input the buffer from the previous time step, the context vector calculated using attention, and the previous output. The new buffer frame is added to the buffer in a FIFO(first in first out) manner.

We achieve speaker dependence by adding a projection of the speaker embedding Z computed earlier to the new buffer frame. The output is generated using No, which takes as input the entire buffer and adding the projection of Z. The output of the network are vocoder features, which when fed into a WORLD vocoder to produce sound.

C.WORLD vocoder

WORLD is a high quality speech synthesis system that also meets real-time processing requirements. It contains three algorithms for obtaining three speech parameters and a synthesis algorithm that takes these parameters as input. Fundamental frequency F0 estimation is done using DIO algorithm. Second, the spectral envelope is estimated with CheapTrick, which uses not only the waveform but also the F0 information. Third, the excitation signal is estimated with PLATINUM and used as an aperiodic parameter. PLATINUM uses the waveform, F0, and spectral envelope information. [8]

DIO F0 estimation algorithm consists of three steps. The first step is to low-pass filter the signal with different cutoff frequencies. If the filtered signal only consists of the fundamental component, it forms a sine wave with a period of T0, which is the fundamental period. Since the target F0 is unknown, many filters with different cutoff frequencies are used in this step. The second step is to calculate the F0 candidates and their reliability in each filtered signal. A signal that consists of only the fundamental component forms a sine wave i.e., the four intervals of the waveform - the positive and negative zero-crossing intervals and peak and dip

intervals, have the same value. Their standard deviation is therefore associated with the reliability measure and their average is defined as an F0 candidate. In the third step, the candidate with the highest reliability is selected.

CheapTrick is based on the idea of pitch synchronous analysis and uses a Hanning window with the length of 3T0. First, the power spectrum is calculated on the basis of the windowed waveform.

$$\int_0^{3T_0} (y(t)w(t))^2\,dt \;=\; 1.125 \int_0^{T_0} y^2(t)dt, \tag{1}$$

$$P_s(\omega) = \frac{3}{2\omega_0} \int_{-\frac{\omega_0}{3}}^{\frac{\omega_0}{3}} P(\omega + \lambda)d\lambda, \tag{2}$$

$$P_l(\omega) = \exp\left(\mathcal{F}\left[l_s(\tau)l_q(\tau)p_s(\tau)\right]\right), \tag{3}$$

$$l_s(\tau) = \frac{\sin(\pi f_0 \tau)}{\pi f_0 \tau}, \tag{4}$$

$$l_q(\tau) = \tilde{q}_0 + 2\tilde{q}_1 \cos\left(\frac{2\pi\tau}{T_0}\right), \tag{5}$$

$$p_s(\tau) = \mathcal{F}^{-1}\left[\log\left(P_s(\omega)\right)\right], \tag{6}$$

The overall power of the windowed waveform is temporarily stabilized. Then, the power spectrum is smoothed with a rectangular window of width 2ω0/3. After that special liftering is carried out where equation 4 and 5 represent special liftering functions for smoothening logarithmic power spectrum and removing time components and for spectral recovery respectively.

$$S_m(\omega) = \exp\left(\mathcal{F}[c_m(\tau)]\right), \tag{7}$$

$$c_m(\tau) = \begin{cases} 2c(\tau) & (\tau > 0) \\ c(\tau) & (\tau = 0) \\ 0 & (\tau < 0) \end{cases}, \tag{8}$$

$$c(\tau) = \mathcal{F}^{-1}[\log(P_l(\omega))], \tag{9}$$

$$x_p(t) = \mathcal{F}^{-1}[X_p(\omega)], \tag{10}$$

$$X_p(\omega) = \frac{X(\omega)}{S_m(\omega)}. \tag{11}$$

PLATINUM windows the waveform by using a window with a length of 2T0. The spectrum of the windowed signal X(ω) is divided by the minimum phase spectrum S m(ω). Then the extracted excitation signal xp(t) is expressed in equations 10 and 11. [8][9]

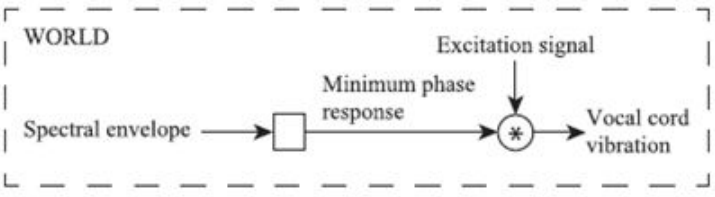

Fig.3. Voice Synthesis using WORLD Vocoder

As shown in Figure 2, an array of output signal with length of output is initialised along with impulse response with length that is half the size of the FFT window. Minimum phase analysis is performed followed by inverse real FFT, getting a time base and removing DC components from the signal obtained. Lastly, periodic and aperiodic responses are calculated for a time to obtain the final synthesized speech signal.

V. TESTING

Selecting test data wisely to evaluate a text to speech system is important. Here a test data was created which includes all possible variations including numerals, dates, address, salutations, tongue twisters, homographs, difficult words, semantically unpredictable words, punctuations and sentences which evaluate prosody. Such data helps in verifying the system specifications, features, and in defining its limitations. Module testing was done for diagnosing the issues that may degrade the performance of specific modules. Our system consists of four main modules: Phonemizer, Context generation, Buffer updation and Output generation. These modules were evaluated over different variations present in the test data.

VI. CONCLUSION

In experiments, single-speaker TTS and multi-speaker TTS along with speaker identification (ID) were tested on the model,which showed that the proposed approach outperforms baselines, namely, Tacotron and Char2wav.[20][2]. Finally, challenging Youtube data with background noise was used to train the model ,which showed promising results.

Pros:

1. It uses relatively simple and less number of parameters by using shallow fully-connected neural networks.
2. Using shifting buffer memory gives a novel solution for memory management..
3. The proposed approach outperforms baselines in several tasks, and the ability to fit to a novel speaker is quite accurate.

Cons:

1.Using the dataset the sentences containing the phoneme ‘zh’ could not be processed.

2.More efficient use of buffer is possible as all 20 buffer elements are not being used at each time instance.

3.Less accuracy for exceptional cases .It can be improved with training the model with huge and precise data but at the cost of more storage and time.

Acknowledgment

The authors acknowledge the support provided by the Department of Electronics and Telecommunications Engineering, Vishwakarma Institute of Technology, Pune, India.

References

[1]Yaniv Taigman, Lior Wolf, Adam Polyak and Eliya Nachmani Facebook AI Research,"Voice Loop: Voice fitting and synthesis via a phonological loop", In Proc. International Conference on Learning Representations (ICLR),1 Feb 2018

[2]Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. Char2wav: End-to-end speech synthesis. In ICLR workshop, 2017.

[3]Sercan O Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, et al. Deep voice: Real-time neural text-to-speech. In Proc. of the 34th International Conference on Machine Learning (ICML), 2017b.

[4] O'Malley, M. H. (1990). Text-to-speech conversion technology. Computer, 23(8), 17–23. doi:10.1109/2.56867

[5]. Ungurean, D. Burileanu, "An advanced NLP framework for high-quality Text-to-Speech synthesis," SpeD 2011, pp. 1-6, Braşov,Romania,18-21 May 2011

[6] Diemo Schwarz. A System for Data-Driven Concatenative Sound Synthesis. Digital Audio Effects (DAFx), Dec 2000, Verona, Italy. pp.97-102. ⟨hal-01161115⟩

[7]KD Bamane, KN Honwadkar,Marathi speech Synthesized Using Unit selection Algorithm, Computer Engineering and Intelligent Systems ISSN, 2011

[8]WORLD: A Vocoder-based High-Quality Speech Synthesis System for Real-Time Applications - Masanori MORISE et. al, July 2016

[9] REFERENCE Manual for Speech Signal Processing Toolkit ver 3.9, December 25, 2015

[10]Robert L Weide. The CMU pronouncing dictionary. URL: http://www. speech. cs. cmu. edu/cgibin/cmudict, 1998.

[11]Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit, 2017.

[12]Zhizheng Wu, Oliver Watts, and Simon King. Merlin: An Open Source Neural Network Speech Synthesis System, pp. 218–223. 9 2016.

[13]Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. Char2wav: End-to-end speech synthesis.

[14]Firoj Alam, Mumit Khan: Bangla Text to Speech using Festival

[15]Ramanpreet Singh, Dharamveer Sharma: An Improved System for Converting Text into Speech for Punjabi Language using eSpeak

[16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves,
Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw
audio. arXiv preprint arXiv:1609.03499, 2016.

[17]Alex Graves. Generating sequences with recurrent neural networks., arXiv:1308.0850v5 [cs.NE] 5 Jun 2014

[18]Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. arXiv preprint arXiv:1410.5401, 2014.

[19]Paul Michel, Okko Rasanen, Roland Thiolliere, Emmanuel Dupoux: Blind Phoneme Segmentation With Temporal Prediction Errors, July 2017

[20]Yuxuan Wang,R. J.Skerry-Ryan,Tacotron:Towards End-to-End Speech Synthesis,INTERSPEECH 2017