Arabic Speech Synthesis using Deep Neural Networks

Aya Hamdy Ali
Language & Speech innovation centre
Global Technical Services (GTS) co.
Cairo, Egypt
aya.hamdy@fcih.net

Mohamed Magdy
Language & Speech innovation centre
Global Technical Services (GTS) co.
Cairo, Egypt
mhmd.magdi@gmail.com

Maher Alfawzy
Language & Speech innovation centre
Global Technical Services (GTS) co.
Cairo, Egypt
maher.alfawzy@hotmail.com

Mikhail Ghaly
Language & Speech innovation centre
Global Technical Services (GTS) co.
Cairo, Egypt
ghalimichael@gmail.com

Hazem Abbas
Computers and Systems Engineering Department
Faculty of Engineering,
Ain Shams University,
Cairo, Egypt
hazem.abbas@eng.asu.edu.eg

***Abstract*— Text-to-speech (TTS) synthesis is a rapidly growing field of research. Deep learning has shown impressive results in speech synthesis and outperformed the older concatenative and parametric methods. In this paper, speech synthesis using deep learning architectures is explored and two models are utilized in an end-to-end Arabic TTS system. The results of the two systems are compared to concatenative TTS system using the Mean Opinion Score (MOS) of the synthesized speech and indicates that deep learning based systems have outperformed the concatenative system when it comes to naturalness and intelligibility; moreover, it reduces system complexity.**

Keywords—text-to-speech, deep learning, recurrent architecture, speech processing

I. INTRODUCTION

There is currently a lot of research carried out in the area of text-to-speech synthesis. This work builds upon the state-of-the-art in neural speech synthesis and attention-based sequence to-sequence learning. In [1], Wang et al., present Tacotron, an end-to-end generative text-to-speech model based on the sequence-to-sequence (seq2seq) model [2] that takes characters as input and outputs raw spectrogram and synthesizes speech directly from spectrogram using Griffin-Lim method [3] as the synthesizer. Given <text, audio> pairs, the model can be trained completely from scratch with random initialization. Tacotron achieves a 3.82 subjective 5-scale mean opinion score on US English, outperforming a production parametric system in terms of naturalness. In [4], van den Oord et.al, presents WaveNet a generative model for generating raw audio waveforms based on the Pixel Convolutional Neural Network PixelCNN architecture [5]. The proposed model is fully probabilistic and autoregressive; it requires conditioning on linguistic features from an existing TTS system so is not fully end-to-end but it is capable of producing audio that is very similar to a human voice. In [6], Sotelo .et.al, preset Char2Wav, an end-to-end model for speech synthesis which can be trained on characters. Char2Wav has two components that needed to be separately pre-trained: the first component is a model based on encoder-decoder model with attention. The encoder is a seq2seq network that accepts text or phonemes as inputs, while the decoder is a recurrent neural network with attention that produces vocoder acoustic features. The second component is a vocoder which generates raw waveform from acoustic features using a Sample RNN neural vocoder [7]. In [8], O. Arik et.al, presents Deep Voice as a text-to-speech system developed using deep neural networks.

Deep Voice lays the groundwork for truly end-to-end neural speech synthesis. The system comprises five major building blocks based on deep neural networks: a segmentation model for locating phoneme boundaries, a grapheme-to phoneme conversion model, a phoneme duration prediction model, a fundamental frequency prediction model, and an audio synthesis model. In [9], Skerry-Ryan et.al, present a speech synthesis architecture based on Tactron [1] and Wavenet [4]. The system is composed of a recurrent sequence-to-sequence feature prediction network that maps character embedding to mel-scale spectrograms, followed by a modified WaveNet model acting as a vocoder to synthesize time domain waveforms from those spectrograms. The nearly fully end-to-end TTS model in previous work is Tacotron introduced models and it also has the significant advantage of being frame-level and thus highly efficient.

For these reasons we utilized these two models and compared them to create the proposed Arabic TTS systems. The rest of the paper is organized as follows. Section II introduces the proposed model along with the needed data preprocessing and method of synthesis. Experimental results and analysis are presented in Section III and finally the paper is concluded in Section IV.

II. PROPOSED TTS ARCHITECTURE

The goal of this work is to build an end-to-end TTS system for Arabic which can generate natural speech. We present an end-to-end TTS system that learns to synthesize speech directly from (text, audio) pairs based on Google's Tacotron. Given <text, audio> pairs, end-to-end TTS systems can be trained using <text, audio> which reduces system complexity and keeps output good quality. Initial results show that an end-to-end platform can provide better results since intonation of the input phonemes are kept in the model to carry the prosodic features to generate the required contour. Thus, the end-to-end approach was adopted instead of the engineering-based Hidden Markov Model (HMM) original formulation. We created Arabic 'NUN' dataset to train a voice with reasonable naturalness.

The pipeline of the proposed system is depicted in Fig. 1 where the undiacritized Arabic text (input) is vowelized by Natural Language Processing NLP components, then the corresponding phonetic transcription is generated by the phonetizer. Afterwards the speech synthesizer based on Deep learning and signal processing techniques

978-1-7281-6535-6/20/$31.00 ©2020 IEEE

produces the equivalent raw spectrogram. Finally, Tacotron generates the synthesized wave (output).

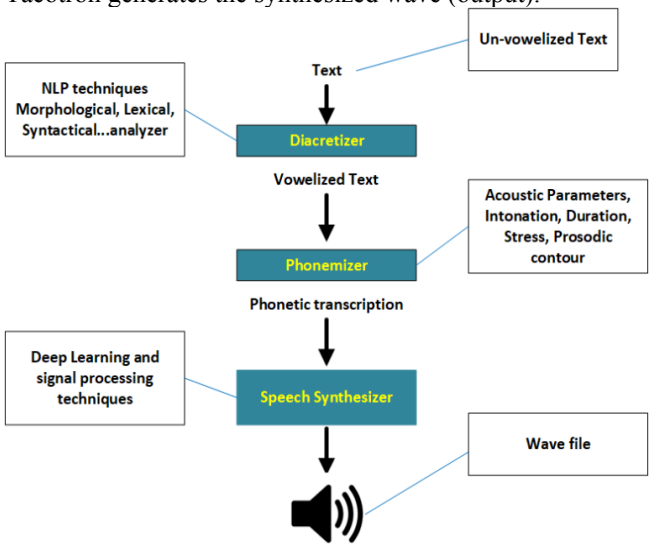

Fig. 1: Pipeline of Arabic TTS components

A. Data preprocessing

1) ***Diactrization****:* Arabic NLP is at least one order of magnitude higher than English and is a must in learning and understanding Arabic for many reasons. Complexity of Arabic NLP is clear on multiple levels: On word level, with high affixation, where one word in Arabic can be translated into an entire English sentence such as:

They will teach both of you	سيعلمونكما	س + يعلم + ون + كما

With the Lack of diacritics and without vowels, one can get up to 10 alternatives:

م ل ك	<--	مَلِكْ	King
		مَلَكْ	Angel
		مُلْكْ	Throne
		مَلَّكَ	Granted
		مِلْكْ	Possession
		مُلِّكَ	Was granted

On the sentence level, in the below sentence, there are 40 alternatives that need a powerful diacritizer to process huge probable alternatives due to morphological [10], Lexical, Syntactic etc... ambiguities:

The man ate the carrots أَكَلَ الرَّجُلُ الْجَزَرَ					
الجزر		الرجل		أكل	
الجَزَر	**The**	الرَجُل	**The**	أكَلَ	**Ate**

الجُزُر الجَزْر	**carrots** The Islands The Ebb	الرجْل	**man** The leg	أَكَّلْ أُكِلَ أَكَّلَ أُكِّل أكِّل	Feed Bored Fed Are all Food

There are several tools and in this paper we utilize Farasa Diactrization API [11] in our proposed system.

2) ***Phonetization****:* Text phonetizer is a critical component in any NLP domain that envisages real system TTS conversion.
Converting from written text into actual sounds is marked by several problems. For Modern Standard Arabic (MSA), these problems are not as severe as they are for English.
However, due to co-articulations, sounds in Arabic can have enormous contextual variability. This requires that a set of rules have to be developed to cover these phonetic variations [12].

Table 1: The Speech Assessment Methods Phonetic Alphabet (SAMPA) for Arabic

Arabic grapheme	Phonemic symbol	Arabic grapheme	Phonemic symbol
Consonants			
ء	/ʔ/	ض	/dˤ/
ب	/b/	ط	/tˤ/
ت	/t/	ظ	/ðˤ/
ث	/θ/	ع	/ʕ/, /ʔ\/
ج	/ʤ/	غ	/ɣ/
ح	/ħ/	ف	/f/
خ	/x/	ق	/q/
د	/d/	ك	/k/
ذ	/ð/	ل	/l/
ر	/r/	م	/m/
ز	/z/	ن	/n/
س	/s/	ه	/h/
ش	/ʃ/	و	/w/
ص	/sˤ/	ي	/j/
Vowels		Diphthongs	
ـَ	/a/		/aj/
ـَا	/aː/		/aw/
ـِ	/i/		
ـِي	/iː/		
ـُ	/u/		
ـُو	/uː/		

Table 2: Standard Arabic vowel system

Tongue position/height	**Front**	**Central**	**Back**
High or closed	**/i/ /iː/ (Unrounded)**		
Low or open		**/a/ /aː/ (Unrounded)**	
High or closed			**/u/ /uː/ (Rounded)**

In this paper, we utilize a graphemes-to-phonemes converter based on SAMPA as in Table 1. We convert Arabic diacritised text to a sequence of phonemes according to the Standard Arabic vowel system in Table 2 and create a pronunciation dictionary from them for alignment using Hidden Markov Model Toolkit (HTK).

B. Speech Synthesis

Synthesized speech is the ultimate production of a TTS system. We previously used the concatenative model but multiple phases in this well-known and traditional approach propagate any error. In this approach, the basic procedures involved in training a set of subword models to generate a prosody model, is based on HMM. As illustrated in Fig. 2, the core process involves the embedded training tool HEREST from HTK.

HEREST uses manually segmented utterances as its source of training data and simultaneously re-estimates the complete set of subword HMMs. For each input utterance, HEREST needs a transcription, i.e. a list of the phones in that utterance. HEREST then joins all of the sub word HMMs corresponding to this phone list to make a single composite HMM [13] that is used to collect the necessary statistics for the re-estimation. When all of the training utterances have been processed, the total set of accumulated statistics are used to re-estimate the parameters of the entire phone HMMs [14]. It is important to emphasize that in this process, no phone boundary information is needed [15] as the transcriptions identify phones sequence in each utterance

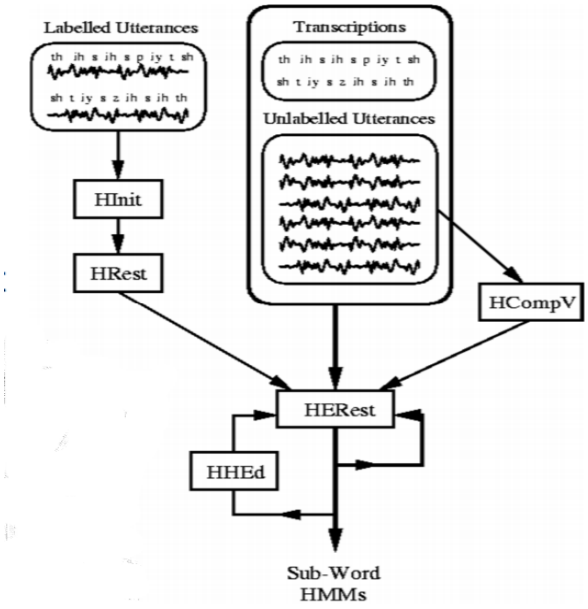

Fig. 2 – HMM based prosodic model [11]

One of the major problems found in building any HMM-based system is that the amount of training data for each model will be variable and is rarely sufficient.

The initialization of a set of phone HMMs prior to embedded re-estimation using HEREST can be achieved by a small set of hand-labelled bootstrap training data to initialize each phone HMM individually [16]. When used in this way, HEREST uses the label information to extract all the segments of speech corresponding to the current phone HMM in order to extract best intonation parameters.

Thus, HMM model quality saturates after a few hours of training data and the resulting outcome of the prosodic model, in spite of being satisfactory, can still be distinguished from human beings. On the other hand, speech synthesis based on Deep learning enhances the quality in a proportional way with the amount of training data. Thus, the merging of both RNN and Long short-term memory (LSTM), produced a quasi-human prosody [17].

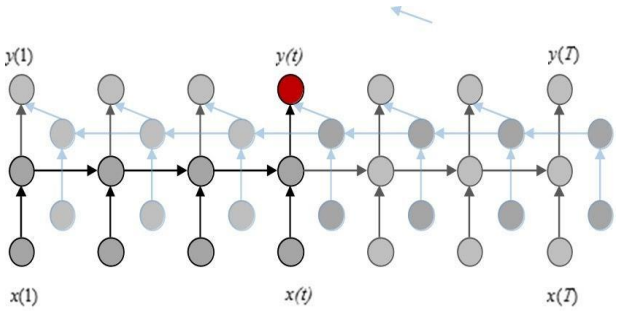

Fig. 3–RNN [22]

As depicted in Fig. 3, RNN connections are between nodes and form a directed graph along a sequence and thus can use their internal state (memory) to process sequences of inputs [18].

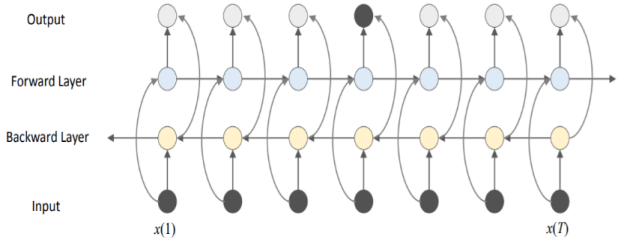

Fig. 4 –LSTM [22]

LSTM blocks are actually building units for RNN layers. It is composed of a cell, an input, an output and forget gates, as shown in Fig. 4. The cell is responsible for "remembering" values over arbitrary time intervals; the word "memory" in LSTM is because the cell is responsible for "remembering" values over time intervals [19].

A Deep learning architecture was selected after applying a similar one for acoustic modeling in speech recognition [20]. This is a replacement of the HMM generation module [21] in the concatenative speech synthesis and will provide, as will be shown below, much better results due to “memory” concept in both RNN and LSTM [22]. Thus, in the new end-to-end approach, our model takes characters as input and outputs raw spectrogram.

In this paper, to synthesize speech, we have utilized RNN-based Seq2Seq model for generating mel spectrogram from text. The architecture is similar to Tacotron 2 [5].

The generated mel spectrogram can either be inverted via iterative algorithms such as Griffin Lim, or through more complicated neural vocoder networks such as a mel spectrogram conditioned Wavenet [2,4]-

WaveNets are a high-quality approach to a” neural

backend". We investigated in this research Tactron and Tactron2 and evaluated them.

In this paper, we implement two modified versions of Tacotron 2 and Tacotron architecture to work with the Arabic language. Tacotron 2 is a modified version of Tacotron where the simpler Griffin-lim algorithm is replaced by WaveNet vocoder. Tactron Model is illustrated in Fig. 5, and the modified model Tactron2 is shown in Fig. 6. Before training, the Arabic data is preprocessed as described in the previous sections then the input text enters the model and the characters are converted into a 512-dimensional character embedding and target spectrograms are computed from the waveforms in the dataset through a short-time Fourier transform (STFT). Spectrograms are also put through a pre-emphasis filter in order to reduce high frequency noise. This filtering turned out to significantly improve the subjective audio quality of waveforms re-synthesized with the synthesizer. The biggest difference is that during waveform generating Tactron is using Grifflim algorithm while Tactron2 is employing WaveNet model for speech synthesis.

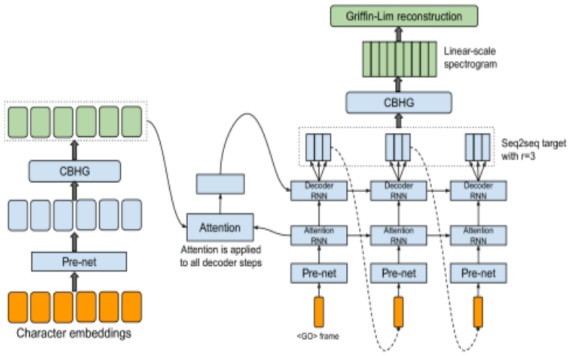

Fig 5. Tacotron model architecture. The model takes

characters as input and outputs the corresponding raw spectrogram, which is then fed to the Griffin-Lim reconstruction algorithm to synthesize speech. [1]

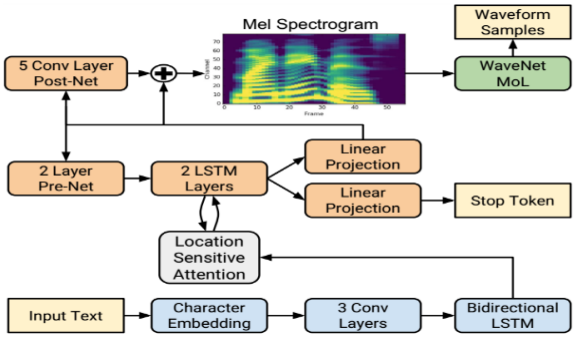

Fig. 6. Tactron2 Model architecture. The model takes

characters as input and outputs the corresponding raw spectrogram, which is then fed to WaveNet Model to synthesize speech. [23]

III. EXPERIMENTAL RESULTS AND ANALYSIS

NUN Dataset:

Synthesizing natural speech requires training on a large number of high-quality speech-transcript pairs, thus we recorded NUN Arabic dataset. The NUN Corpus is designed to provide speech data for text to speech. The NUN Corpus contains 4.5 hours of high-quality recordings of only one speaker. Speaker is reading About 5000 phonetically rich sentences covering all Arabic phones and DiPhones. The NUN corpus includes sentences as well as a 16-bit, 48kHz speech waveform file for each sentence. Corpus design was a joint effort among engineers and computational linguists.

Training Setup:

In the case of Tactron, the model is trained for 1000k steps with batch size 32 on a single GPU. We use the Adam optimizer [24] with values β1 = 0:9; β2 = 0:999 and learning rate to 0.001 in Tacotron and Tacotron 2. Seven test sentences are synthesized using saved models from several points in training history to select best performance checkpoint on both naturalness and intelligibility.

In case of Tactron 2, the model is trained into two separate steps: first training the feature prediction network on its own for 1000k steps with batch size 32 on a single GPU, followed by training a WaveNet independently on the outputs generated by the first network for 1.9M steps with batch size 8. Figures 7 and 8 illustrate the loss curve when training Tactron and Tacotron 2 models on our Arabic dataset.

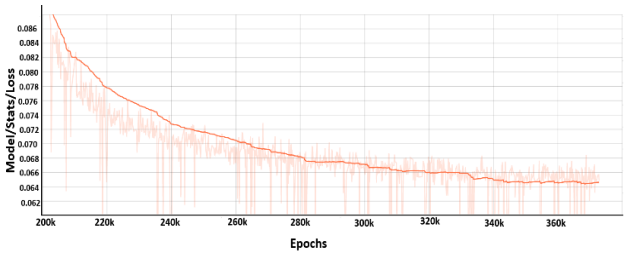

Fig. 7. Tacotron 1 Training loss

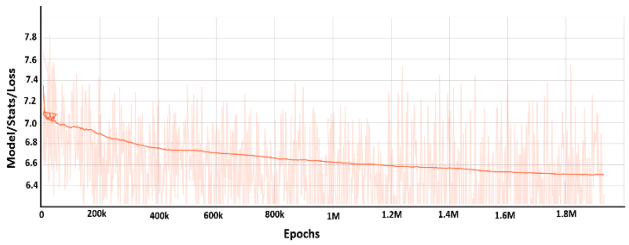

Fig. 8. Tacotron 2 Training loss

Analysis:

The quality of the synthesized speech is evaluated through a subjective test. We randomly selected 20 fixed examples from the test set of our internal dataset as the evaluation set.

Audio generated from Tacotron and Tacotron 2 on this set where each sample is rated by at least 8 raters on a scale from 1 to 5 with 0.5 point increments, from which a

subjective MOS is calculated. Each evaluation is conducted independently from each other, so the outputs of two different models are not directly compared when raters assign a score to them. The two systems are built on the same dataset which should minimize the risk for biases towards certain types of voices.

To evaluate both models against Concatenative HMM, we used MOS like typical TTS system evaluations. To get this score, samples of a TTS output are given to native-speakers and rated on a score from 1 (Bad) to 5 (Excellent) according to speech main criteria's: Intelligibility, Naturalness, Pleasance, Smoothness and Prosody. The subjects were asked to rate the output and the MOS is then computed as the arithmetic mean of these scores:

Where R are human ratings for a TTS sample by N people.

For the 3 TTS models we collected and generated MOS score from 50 people with a normal distribution of age, gender and education to represent a neutral sample of Arabic audience.

The subjective listening tests were blind: 10 sentences not included in the training data were used for the 50 testers with Concatenative, Tacotron 1 and Tacotron 2, respectively i.e., speech samples have the same text synthesized by the different models. As shown in Table 3, Tacotron 2 achieves an MOS of 4.38, while Tacotron 1 gives 4.01 which outperforms the concatenative system 3.89 and represents promising results.

Table 3: MOS results

Arabic TTS model	MOS
Concatenative with HMM	3.89
Tacotron 1	4.01
Tacotron 2	4.38

IV. CONCLUSION

In this paper, we presented a Text to Speech using Deep Learning latest techniques for generation of MSA text. Our experimental results showed the efficacy of the proposed method, in comparison to a conventional concatenative HMM-based approach concerning the output quality. Future work will include the use of sophisticated optimization approaches, to improve the synthesis generation time.

REFERENCES

[1] Wang, Yuxuan, Skerry-Ryan, RJ, Stanton, Daisy, Wu, Yonghui, Weiss, Ron J., Jaitly, Navdeep, Yang, Zongheng, Xiao, Ying, Chen, Zhifeng, Bengio, Samy, Le, Quoc, Agiomy Giannakis, Yannis, Clark, Rob, and Saurus, Rif A. Tacotron: Towards end-to-end speech synthesis. In Proceedings of Interspeech, August 2017a.

[2] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in neural information processing systems, 2014, pp. 3104–3112.

[3] D. Griffin and J. Lim, "Signal estimation from modified short time Fourier transform," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, no. 2, pp. 236–243, 1984.

[4] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," arXiv:1609.03499v2, 2016.

[5] van den Oord, Aaron, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks¨, arXiv:1601.06759, 2016a.

[6] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in ICLR2017 workshop submission, 2017.

[7] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," arXiv preprint arXiv:1612.07837, 2016

[8] Sercan O. Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, Shubho Sengupta, Mohammad Shoeybi, "Deep Voice: Real-time Neural Text-to-Speech", arXiv:1702.07825v2 , 2017

[9] Skerry-Ryan, R. J., Battenberg, E., Xiao, Y., Wang, Y., Stanton, D., Shor, J., ... & Saurous, R. A. (2018). Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. arXiv preprint arXiv:1803.09047.

[10] Beesley, Kenneth, "Arabic Finite-State Morphological Analysis and Generation. COLING-96, 1996".

[11] Darwish, K., Mubarak, H., & Abdelali, A. (2017, April). Arabic diacritization: Stats, rules, and hacks. In Proceedings of the Third Arabic Natural Language Processing Workshop (pp. 9-17).

[12] Yousif A. El-Imam "Phonetization of Arabic Rules and algorithms," Computer Speech & Language, v. 18, pp. 339-373, 2004.

[13] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, T. Kitamura, Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis, Proc. of Eurospeech, pp.2347-2350, 1999.

[14] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayahsi, T. Kitamura, "Mixed excitation for HMM-based speech synthesis", in Proc. EuroSpeech, 2001.

[15] K. Tokuda, T. Mausko, N. Miyazaki, T. Kobayashi, Multi-space probability distribution HMM, IEICE

Trans. Inf. & Syst., vol. E85-D, no.3, pp.455-464, 2002.
[16] T. Toda and K, K. Tokuda, Speech Parameter Generation Algorithm Considering Global Variance for HMM-Based Speech Synthesis, in Proc. Eurospeech, 2005.
[17] R. Fernandez, A. Rendel, B. Ramabhadran, R. Hoory, "Prosody Contour Prediction with Long Short-Term Memory, Bi-Directional Deep Recurrent Neural Networks", In. Proc. Interspeech 2014.
[18] S. Mike, K. Paliwal. "Bidirectional recurrent neural networks." IEEE Transactions on Signal Processing, vol.45, no.11, pp.2673-2681, 1997.
[19] H. Sepp, S. Jürgen, "Long short-term memory." Neural computation, vol.9, no.8, pp. 1735-1780, 1997.
[20] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," Signal Processing Magazine, IEEE, vol. 29, no. 6, pp. 82–97, 2012.
[21] Steve Young et al. The HTK Book. Version 3.4, March 2006.
[22] Deep Learning for Speech Generation and Synthesis, Yao Qian, Frank K. Soong, Speech Group MS Research Asia, September 13, 2014
[23] Yuan (Gary) Wang , "Deep Text-to-Speech System with Seq2Seq Model", , arXiv:1903.07398v1, 2019.
[24] Diederik P. Kingma, Jimmy Ba, "Adam: A Method for Stochastic Optimization", arXiv:1412.6980v9, 2014.