A Fully Neural Tunisian Arabic TTS System

Moez Ben HajHmida MOEZ.BENHAJHMIDA@ENIT.UTM.TN
National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia
Hatem Haddad HADDAD.HATEM@GMAIL.COM
RIADI Laboratory, National School for Computer Sciences, University of Manouba, Tunis, Tunisia
Aymen Ben El Haj Mabrouk BHMMA18@GMAIL.COM
National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia

Abstract

The discipline of Text-To-Speech (TTS) focuses on the artificial generation of spoken language from text, a technology increasingly vital for voice-based applications. Recognizing the rising demand for realistic computer-generated speech, especially for under-represented accents and dialects, this research is driven by the goal of constructing a high-quality Tunisian Arabic TTS system. Highlighting the underdeveloped state of advanced natural language processing technologies like TTS in Tunisia, this paper introduces our work on recording a dedicated Tunisian female Arabic speech dataset. Furthermore, we present an end-to-end deep learning TTS system built upon a deep neural network architecture. A subjective evaluation using Mean Opinion Score (MOS) was conducted, comparing our approach to end-to-end generative and concatenative models. The results of this evaluation indicate that our proposed system outperforms both baselines in terms of both naturalness and intelligibility.

Keywords: Text-To-Speech, Dialects, Deep Neural Networks, Tacotron 2, WaveRNN, Griffin-Lim Algorithm

1. Introduction

Tunisian dialect is a variant of the Arabic language spoken in Tunisia with various distinct features, such as its own vocabulary, grammar, and pronunciation. The work on speech synthesis for underrepresented languages, such as the Tunisian dialect, is important. One of the main reasons is that it helps to preserve and promote linguistic diversity. Additionally, speech synthesis in underrepresented languages can help to improve communication and access to information for people who speak those languages, including providing education, healthcare, and other services. Furthermore, it can also help bridge the digital divide and promote economic development in regions where these languages are spoken.

Tunisian Dialect (TD) (Haddad et al., 2023) and Modern Standard Arabic (MSA) are distinct variations of Arabic, sharing commonalities but also significant differences. TD incorporates numerous loanwords from Berber, French, Turkish, and Italian, which are not found in MSA (Messaoudi et al., 2021). It also features a unique accent, intonation, and pronunciation variations. Grammatically, TD includes sounds like "gu", "v", and "p" alongside unique structures and conjugations not present in MSA. In summary, TD has its own phonetics, lexicon, and morphological structures, as shown in Table 1.

Traditionally, concatenative synthesis (Bhushan et al., 2024) constituted the state-of-the-art in Text-to-Speech (TTS). However, this method necessitated substantial domain-specific knowledge and manual effort for the precise alignment of linguistic and acoustic

Table 1: Examples of Tunisian sentences with their MSA and English translation.

Tunisian	MSA	English
قداش الوقت بربي؟	كم الساعة لو سمحت؟	What time is it please?
ما نحبش برشا الغلة	لا أحب الغلال كثيرا	I don't like fruits
تنجم تسلفني فلوس؟	هل تستطيع أن تقرضني مال؟	Can you lend money?

characteristics. In recent years, Deep Neural Network (DNN) have dramatically reshaped the landscape of speech synthesis and numerous natural language processing applications.

DNN based systems have become more and more popular for TTS. A DNN TTS system is generally composed of two parts: an encoder/decoder component which predicts a spectrogram (Koenig et al., 1946) or a mel-spectrogram (Davis and Mermelstein, 1980) from a text input, and a vocoder component which converts a spectrogram to a time-domain representation. Tacotron (Wang et al., 2017) is an end-to-end generative TTS model that takes a character sequence as input and outputs the corresponding spectrogram. Tacotron uses the Griffin-Lim algorithm (Griffin and Lim, 1983) for phase estimation, followed by an inverse short-time Fourier transform. The Tacotron 2 (Shen et al., 2018) model is a fully neural Text-to-Speech system that leverages a sequence-to-sequence recurrent neural network with an attention mechanism to predict mel-spectrogram features, which are subsequently used by a modified WaveNet vocoder (Oord et al., 2016) to synthesize the audio waveform. Inspired by the success of the Transformer network in neural machine translation, TransformerTTS (Li et al., 2019) is an architecture based on Tacotron2 and Transformers. Neural network-based TTS has outperformed conventional concatenative and statistical parametric approaches in terms of speech quality (Ren et al., 2019).

This study outlines the process of creating a Tunisian Arabic TTS system leveraging the Tacotron 2 architecture. We explore the impact of different vocoders, specifically Griffin-Lim and WaveRNN, on the synthesized speech quality. Furthermore, we provide a comparative analysis against other TTS methods, including concatenative and DNN-based systems.

We structure the paper as follows. Section 2 describes related state-of-the-art approaches. Section 3 introduces the architecture of the system used in this study. Section 4 describes the training dataset and illustrates the evaluation methods. Finally, Section 5 shows conclusions and future work.

2. Related Work

Tacotron 2 (Shen et al., 2018) is a state-of-the-art neural network architecture for speech synthesis developed by Google. It is designed to generate human-like speech from textual input. The system consists of two main components: a sequence-to-sequence model that maps text sequences to mel-spectrograms, and a modified WaveNet vocoder that generates time-domain waveforms from these spectrograms. Tacotron 2 improves upon its predecessors by employing a recurrent neural network (RNN) with attention mechanisms to capture long-range dependencies and produce more accurate spectrogram predictions. The system is noted for its ability to produce natural and expressive speech, accurately capturing

nuances in tone, pronunciation, and inflection, making it highly effective for applications in voice assistants, audiobooks, and other areas requiring high-quality synthesized speech. The effectiveness of Tacotron 2 on speech synthesis for English, Chinese, and other widely spoken languages motivated researchers to utilize Tacotron 2 for Arabic speech synthesis.

In Hussain et al. (2020), authors presented a German–Arabic speech-to-speech translation system deployed in the context of interpretation during psychiatric, and diagnostic interviews. One of the components of this system is the Arabic speech synthesis. The team trained a Tacotron 2 model on Nawar Halabi's Arabic Dataset [1] and used a pretrained WaveGlow model (Prenger et al., 2019) which predicts a raw waveform from a mel-spectrogram. At a first attempt, authors synthesized the audio from Arabic characters without diacritics. They employed a sub-word tokenization method, byte-pair encoding (BPE) (Sennrich et al., 2016), but the model did not converge. So, they developed and used a diacritization component. They ended by feeding the Tacotron 2 with Arabic character sequences with diacritics converted in Buckwalter format. The resulting speech sounds natural, as reported by authors. When synthesizing for only one word, the model did not terminate and reached the maximum decoding steps. Human evaluators judged that most of the synthesized audios were very close to human speech.

In Ali et al. (2020), the authors compared the Tacotron and Tacotron 2 models with a concatenative HMM-based model for speech synthesis from MSA text. They proposed a pipeline that diacritizes the input text, employs a phonetizer to transform the diactirized text into sound, then generates the audio speech with each of the three models. They trained the three models on the NUN Arabic Text-to-Speech dataset. The NUN dataset comprises 4.5 hours of audio recordings from a single speaker, covering 5,000 sentences. Authors report that Tacotron performs better than the concatenative HMM-based model and Tacotron 2 outperforms them all.

In Fahmy et al. (2020), authors fine-tuned a pretrained Tacotron 2 English model on Nawar Halabi's Arabic Dataset to produce an MSA speech synthesis. Authors utilized Tacotron 2 to generate mel-spectrograms from Arabic diacritic text as an intermediate feature representation. Then, they used a pretrained WaveGlow model to produce the wave audios. Similarly to Ali et al. (2020) work, Fahmy et al. showed that Tacotron with Griffin-Lim performs better than concatenative HMM-based model and Tacotron 2 with WaveGlow generates the most natural audios.

3. Proposed system

A key advantage of end-to-end neural network architectures in speech synthesis, when compared to conventional methodologies, lies in their ability to mitigate the necessity for extensive domain-specific knowledge, laborious manual feature engineering, and substantial human annotation. The general architecture of our proposed speech synthesis system, as shown in Figure 1, consists of several main components:

1. Text pre-processing module: It takes text input and prepares it for the model through operations such as tokenization, punctuation removal, and text normalization.

1. Arabic Speech Corpus Homepage. http://en.arabicspeechcorpus.com/

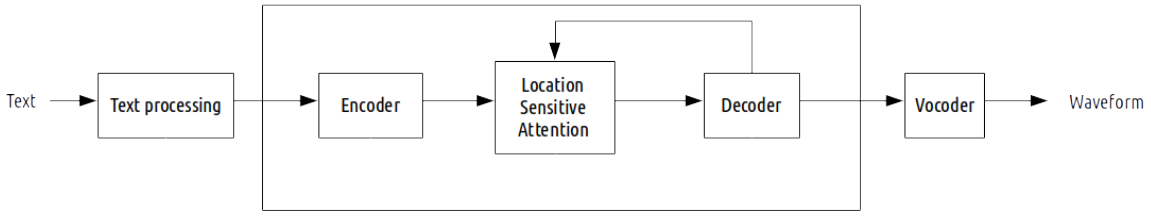

Figure 1: General architecture of the proposed system.

2. Text-to-phoneme conversion (Phonemizer): This converts the pre-processed text into a phoneme sequence, which is vital for achieving natural-sounding speech with correct rhythms and intonation.

3. Encoder module: This module processes the phoneme sequence to generate feature representations or embeddings that capture the text's meaning.

4. A decoder module, which takes in the encoded text and generates a mel-spectrogram, which is a representation of the spectral content of speech.

5. Vocoder module: Finally, the vocoder synthesizes the audio signal from the generated mel-spectrogram.

3.1. Text processing

Our system takes as input diacritized text. We apply a phonemizer (Taylor and Al-Sabbagh, 2010) to convert of the text to phoneme representation using The International Phonetic Alphabet (IPA) (see Table 2). The phonemizer analyzes the input text and generates a sequence of phonemes that match the natural rhythms and intonations of human speech. We utilize the eSpeak[2] phonemizer which uses a set of rules and dictionaries to perform the text-to-phoneme conversion. The rules take into account the context of the word and the surrounding words to generate the correct phonemes. The dictionaries contain information about the pronunciation of words, such as the stress patterns and the phoneme sequences. To support Tunisian dialect accent we enriched the eSpeak Arabic dictionary with the "gu", "v", and "p" sounds.

3.2. Encoder

We applied a sequence-to-sequence network based on Tacotron 2. There are a number of variants in this model. One variant uses linear spectrograms and another uses mel-scale spectrograms and reconstructs them using the vocoder. The sequence-to-sequence prediction model includes different neural networks, as described in Figure 2. The two main networks are the encoder and the decoder with attention. Character embedding is the forefront of the encoder. In our proposed model, character embeddings of size 512 are

2. https://espeak.sourceforge.net/

Table 2: Text to phonemes conversion.

Original text	Phonemized text
قَصْرُ	qasɹu
إِبْتَكَرَ طَائِرَةَ	ʔibtakaɹa taːʔiɹata
إِسعَافٌ	ʔisʕaːfin

applied for handling the phonemized text. The input characters are passed through a stack of 3 convolutional layers each of them followed by batch normalization and ReLU activation function. The encoded features are generated by passing the final convolutional layer's output through a single bi-directional LSTM layer with 256 units in each direction (forward and backward). Following the original Tacotron 2 design, we utilize a location-sensitive attention network. This network computes location features using 32 1D convolution filters of size 31 applied to the previous attention weights.

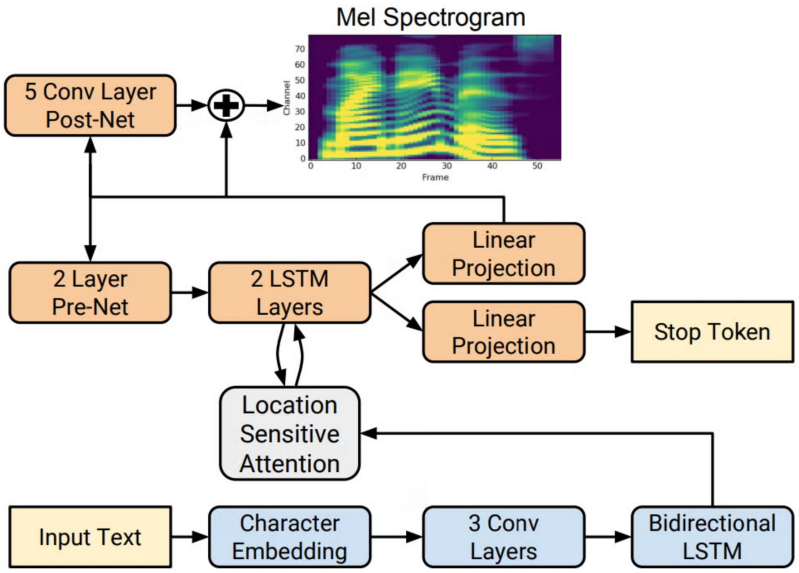

Figure 2: Tacotron 2 - Mel-Spectrogram Prediction Network.

3.3. Decoder

The decoder is an auto-regressive recurrent neural network which predicts a mel-spectrogram from the encoded input sequence one frame at a time. The prediction from the previous time step is first passed through a small pre-net containing 2 fully connected layers of 256 hidden ReLU units. The pre-net output and attention context vector are concatenated and passed through a stack of 2 uni-directional LSTM layers with 1024 units. The concatenation of the LSTM output and the attention context vector is projected through a linear transform to

predict the target mel-spectrogram frame. Finally, the predicted mel-spectrogram is passed through a 5-layer convolutional post-net which predicts a residual to add to the prediction to improve the overall reconstruction. The concatenation of decoder LSTM output and the attention context is projected down to a scalar and passed through a sigmoid activation to predict the probability that the output sequence has completed. This stop token prediction is used during inference to allow the model to dynamically determine when to terminate generation instead of always generating a fixed duration. Specifically, generation completes at the first frame for which this probability exceeds a threshold of 0.5. The convolutional layers in the network are regularized using dropout with probability 0.5, and LSTM layers are regularized using zoneout with probability 0.1.

3.4. Vocoders

A vocoder is a neural network or an algorithm that is used to convert a representation of speech such as a Mel-spectrogram into a speech waveform that can be played as audio. It essentially converts the output of the decoder model into a sound that can be heard by human ears.

3.4.1. Griffin-Lim Algorithm

The Griffin-Lim algorithm (Griffin and Lim, 1983) is an algorithm for reconstructing the time-domain waveform of a signal from its magnitude spectrogram. It is an iterative algorithm that starts with an initial estimate of the signal, and then repeatedly applies a phase reconstruction step and a magnitude projection step until it converges to a satisfactory solution.

The Griffin-Lim algorithm begins by initializing the signal with a random complex-valued signal with the same length as the original signal. Then, it computes the magnitude spectrogram of this initial estimate, which is a representation of the signal in the frequency domain that only contains information about the amplitude of each frequency component. The algorithm then applies an inverse Short-Time Fourier Transform (STFT) (Griffin and Lim, 1983) to the magnitude spectrogram to obtain a new estimate of the time-domain signal. This new estimate may not match the original signal's phase, so the algorithm applies a phase reconstruction step to the new estimate, which involves computing the phase of the original signal's STFT, and combining it with the magnitude of the new estimate's STFT. The new estimate is then projected back to the magnitude spectrogram of the original signal. The process of phase reconstruction, inverse STFT and projection is repeated until the algorithm converges to a satisfactory solution. It is computationally efficient and does not require prior knowledge of the original signal's phase.

3.4.2. WaveRNN

WaveRNN is a type of autoregressive model that is trained to generate audio samples one at a time, conditioned on the previous samples. The model consists of an encoder and a decoder. The encoder converts the input audio signal into a compact representation (latent code) and the decoder generates the output audio signal from the latent code.

The encoder employs convolutional layers for feature extraction from the audio, a downsampling layer for resolution reduction, and fully connected layers to project the features into a lower-dimensional latent space.

The decoder consists of several layers of upsampling and LSTM layers. The upsampling layers are used to increase the resolution of the audio data and the LSTM layers are used to model the temporal dependencies in the audio. The spatial resolution increase required for upsampling is frequently performed with transposed convolution layers. Temporal dependencies within the audio signal are modeled using Long Short-Term Memory (LSTM) layers. Each LSTM cell in the sequence receives the previously generated sample and the preceding hidden state, producing an updated hidden state and the subsequent output sample.

4. Experimental Results

4.1. Training Data

First of all, we used the text included in Nawar Halabi's Arabic Dataset[3] . This speech corpus contains (text, audio) pairs. As some sentences are too long, we split them out in a way to not exceed 178 characters by sentence. We transferred 20% of the sentences to the Tunisian Dialect. Our audio recordings feature a Tunisian female voice actor, captured in a professional studio. The sentences were voiced by the voice actor in a native Tunisian accent. Audios are saved at a sampling rate of 48 kHz. The length of each audio is between 5 seconds and 15 seconds. In this way, we recreate the Arabic speech corpus with a Tunisian accent, and the final speech contains over 1000 (text, audio) pairs.

4.2. Training setup

During training, the model is presented with an audio sample, and the goal is to generate the next sample of the audio based on the previous samples. The model is trained to minimize the difference between the generated samples and the target samples using a loss function such as mean squared error. During inference, the model takes an initial seed audio and generates the next sample, it then uses the generated sample along with the seed as input to generate the next sample and so on, until the desired output is generated. The model was trained for 150 000 training steps and the Hyper-parameters are defined in Table 3. Training the model required 72 hours on an Nvidia GeForce RTX 2080 GPU.

4.3. Evaluation

To evaluate our proposed Text-to-Speech (TTS) system, we carried out a Mean Opinion Score (MOS) subjective evaluation. In this test, 25 participants listened to randomly selected synthesized audio segments and rated their quality on a five-point scale: 1 (Bad), 2 (Poor), 3 (Fair), 4 (Good), and 5 (Excellent).

Table 4 presents a comparative analysis of our proposed TTS architectures along with the sample results reported in Ali et al. (2020) and Fahmy et al. (2020). A detailed examination of this table reveals that the Tacotron 2 architecture demonstrates superior performance compared to the baseline Tacotron model and the concatenative Hidden Markov

3. http://en.arabicspeechcorpus.com/

Table 3: Mel-Spectrogram Prediction Network Hyper-parameters.

Audios Hyper-parameters	
Sample Rate	48 kHz
FFT size	1024
Window length	1024
Hop length	256
Model Hyper-parameters	
Encoder input dimension	512
Decoder input dimension	512
Decoder output dimension	80
Optimization Hyper-parameters	
Optimizer	RAdam (Liu et al., 2019)
Weight decay	1e-6

Model (HMM). Specifically, the evaluation metrics presented in the table indicate a clear advantage for Tacotron 2 in terms of MOS. Furthermore, our vocoder comparison within the Tacotron 2 framework highlights WaveRNN as the top-performing vocoder, achieving the highest quality synthesized speech according to MOS scores. In contrast, while the Griffin-Lim algorithm emerges as the fastest vocoder in terms of synthesis speed, this efficiency comes at the cost of audio quality, as evidenced by its lower scores in MOS.

Table 4: MOS performance for different system architectures.

Model	**MOS**
Concatenative with HMM	3.89
Tacotron with Griffin-Lim	4.02
Tacotron2 with Griffin-Lim (proposed)	4.11
Tacotron2 with Waveglow	4.21
Tacotron2 with WaveNet	4.38
Tacotron2 with WaveRNN (proposed)	4.39

5. Conclusion

This paper presents a novel Tunisian Arabic TTS system built using Tacotron 2's architecture and comparing Griffin-Lim and WaveRNN vocoders. A key contribution is a new Tunisian Arabic audio dataset. Our system outperformed existing concatenative and DNN methods, highlighting the necessity of a Tunisian-specific phonemizer over the standard MSA one. Future work includes integrating automatic diacritization for a larger corpus and developing a DNN-based Tunisian phonemizer to further enhance speech quality.

References

A. Ali, M. Magdy, M. Alfawzy, M. Ghaly, and H. Abbas. Arabic speech synthesis using deep neural networks. In *ICCSPA*, pages 1–6. IEEE, 2020. ISBN 978-1-7281-6535-6. URL http://dblp.uni-trier.de/db/conf/iccspa/iccspa2020.html#AliMAGA20.

U. Bhushan, K. Malipatil, V. Vishruth Patil, V. Anilkumar, S. Ananya, and K. P. Bharath. Hmm and concatenative synthesis based text-to-speech synthesis. In *2024 First International Conference on Software, Systems and Information Technology (SSITCON)*, pages 1–6, 2024. doi: 10.1109/SSITCON62437.2024.10797055.

S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken se. 1980. URL https://api.semanticscholar.org/CorpusID:9049364.

F. K. Fahmy, M. I. Khalil, and H. M. Abbas. A transfer learning end-to-end arabic text-to-speech (tts) deep architecture. In F-P. Schilling and T. Stadelmann, editors, *ANNPR*, volume 12294 of *Lecture Notes in Computer Science*, pages 266–277. Springer, 2020. ISBN 978-3-030-58309-5. URL http://dblp.uni-trier.de/db/conf/annpr/annpr2020.html#FahmyKA20.

D. W. Griffin and J. S. Lim. Signal estimation from modified short-time fourier transform. In *ICASSP*, pages 804–807. IEEE, 1983. URL http://dblp.uni-trier.de/db/conf/icassp/icassp1983.html#GriffinL83.

H. Haddad, A. Cheikh Rouhou, A. Messaoudi, A. Korched, C. Fourati, A. Sellami, M. Ben Hajhmida, and F. Ghriss. Tunbert: Pretraining bert for tunisian dialect understanding. *SN Computer Science*, 4, 02 2023. doi: 10.1007/s42979-022-01541-y.

J. Hussain, M. Mediani, M. Behr, M. Chéragui, S. Stüker, and A. Waibel. German-arabic speech-to-speech translation for psychiatric diagnosis. In Imed Zitouni, Muhammad Abdul-Mageed, Houda Bouamor, Fethi Bougares, Mahmoud El-Haj, Nadi Tomeh, and Wajdi Zaghouani, editors, *WANLP@COLING*, pages 1–11. Association for Computational Linguistics, 2020. ISBN 978-1-952148-38-5. URL http://dblp.uni-trier.de/db/conf/wanlp/wanlp2020.html#HussainMBCSW20.

W. Koenig, H. K. Dunn, and L. Y. Lacy. The sound spectrograph. *The Journal of the AcoustH. K. ical Society of America*, 18(1):19–49, 07 1946. ISSN 0001-4966. doi: 10.1121/1.1916342. URL https://doi.org/10.1121/1.1916342.

N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu. Neural speech synthesis with transformer network. In *AAAI*, pages 6706–6713. AAAI Press, 2019. ISBN 978-1-57735-809-1. URL http://dblp.uni-trier.de/db/conf/aaai/aaai2019.html#Li0LZL19.

L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate and beyond. *ArXiv*, abs/1908.03265, 2019. URL https://api.semanticscholar.org/CorpusID:199528271.

A. Messaoudi, H. Haddad, M. Ben HajHmida, C. Fourati, and A. Ben Hamida. Learning word representations for tunisian sentiment analysis. *Pattern Recognition and Artificial Intelligence*, 1322:329, 2021.

A. 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. 2016. URL http://arxiv.org/abs/1609.03499. cite arxiv:1609.03499.

R. Prenger, R. Valle, and B. Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP*, pages 3617–3621. IEEE, 2019. ISBN 978-1-4799-8131-1. URL http://dblp.uni-trier.de/db/conf/icassp/icassp2019.html#PrengerVC19.

Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu. Fastspeech: Fast, robust and controllable text to speech. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Emily B. Fox, and Roman Garnett, editors, *NeurIPS*, pages 3165–3174, 2019. URL http://dblp.uni-trier.de/db/conf/nips/nips2019.html#RenRTQZZL19.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *ICASSP*, pages 4779–4783. IEEE, 2018. ISBN 978-1-5386-4658-8. URL http://dblp.uni-trier.de/db/conf/icassp/icassp2018.html#ShenPWSJYCZWRSA18.

P. Taylor and R. Al-Sabbagh. Taylor, paul. 2009. text-to-speech synthesis. cambridge: Cambridge university press. 2010. URL https://api.semanticscholar.org/CorpusID:264625186.

Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. Saurous. Tacotron: Towards end-to-end speech synthesis. In Francisco Lacerda, editor, *INTERSPEECH*, pages 4006–4010. ISCA, 2017. URL http://dblp.uni-trier.de/db/conf/interspeech/interspeech2017.html#WangSSWWJYXCBLA17.