Université de Liège

Faculté des Sciences Appliquées

Automatic Multispeaker Voice Cloning

Author:
Corentin Jemine

Supervisor:
Prof. Gilles Louppe

Academic year 2018 - 2019

Graduation studies conducted for obtaining the Master's degree in Data Science by Corentin Jemine

Abstract

Recent advances in deep learning have shown impressive results in the domain of text-to-speech. To this end, a deep neural network is usually trained using a corpus of several hours of professionally recorded speech from a single speaker. Giving a new voice to such a model is highly expensive, as it requires recording a new dataset and retraining the model. A recent research introduced a three-stage pipeline that allows to clone a voice unseen during training from only a few seconds of reference speech, and without retraining the model. The authors share remarkably natural-sounding results, but provide no implementation. We reproduce this framework and open-source the first public implementation of it. We adapt the framework with a newer vocoder model, so as to make it run in real-time.

Contents

1 Introduction

Deep learning models have become predominant in many fields of applied machine learning. Text-to-speech (TTS), the process of synthesizing artificial speech from a text prompt, is no exception. Deep models that would produce more natural-sounding speech than the traditional concatenative approaches begun appearing in 2016. Much of the research focus has been since gathered around making these deep models more efficient, more natural, or training them in an end-to-end fashion. Inference has come from being hundreds of times slower than real-time on GPU (van den Oord et al., 2016) to possible in real-time on a mobile CPU (Kalchbrenner et al., 2018a). As for the quality of the generated speech, Shen et al. (2017) demonstrate near human naturalness. Interestingly, speech naturalness is best rated with subjective metrics; and comparison with actual human speech leads to the conclusion that there might be such a thing as "speech more natural than human speech". In fact, Shirali-Shahreza and Penn (2018) argue that the human naturalness threshold has already been crossed.

Datasets of professionally recorded speech are a scarce resource. Synthesizing a natural voice with a correct pronunciation, lively intonation and a minimum of background noise requires training data with the same qualities. Furthermore, data efficiency often remains one of the shortcomings of deep learning. Training a common text-to-speech model such as Tacotron (Wang et al., 2017) typically requires tens of hours of speech. Yet the ability of generating speech with any voice is attractive for a range of applications be they useful or merely a matter of customization. Research has led to frameworks for voice conversion and voice cloning. They differ in that voice conversion is a form of style transfer on a speech segment from a voice to another, whereas voice cloning consists in capturing the voice of a speaker to perform text-to-speech on arbitrary inputs.

While the complete training of a single-speaker TTS model is technically a form of voice cloning, the interest rather lies in creating a fixed model that is able to incorporate newer voices with little data. The common approach is to condition a TTS model trained to generalize to new speakers on an embedding of the voice to clone (Arik et al., 2017a, 2018; Jia et al., 2018). The embedding is low-dimensional and derived by a speaker encoder model that takes reference speech as input. This approach is typically more data efficient than training a separate TTS model for each speaker, in addition to being orders of magnitude faster and less computationally expensive. Interestingly, there is a large discrepancy between the duration of reference speech needed to clone a voice among the different methods, ranging from half an hour per speaker to only a few seconds.

Our objective is to achieve a powerful form of voice cloning. The resulting framework must be able to operate in a zero-shot setting, that is, for speakers unseen during training. It should incorporate a speaker's voice with only a few seconds of reference speech. These desired results are shown to be fulfilled by (Jia et al., 2018).

Their results are impressive[1], but not backed by any public implementation. We reproduce their framework and make our implementation open-source[2]. In addition, we integrate the model of (Kalchbrenner et al., 2018a) in the framework to make it run in real-time, i.e. to generate speech in a time shorter or equal to the duration of the produced speech. add a word about our results

The structure of this document goes as follows. We begin with a short introduction on TTS methods that involve machine learning. Follows a review of the evolution of the state of the art for TTS and for voice cloning. We then present the work of (Jia et al., 2018) as is, without mention of our own implementation. The remainder of this document focuses on the three individual parts of the framework, and on our implementation of the whole.

2 Statistical parametric speech synthesis

Statistical parametric speech synthesis (SPSS) refers to a group of data-driven TTS methods that emerged in the late 90s. In SPSS, the relationship between the features computed on the input text and the output acoustic features is learned by a statistical generative model (called the acoustic model). A complete SPSS framework thus also includes a pipeline to extract features from the text to synthesize, as well as a system able to reconstruct an audio waveform from the acoustic features produced by the acoustic model (such a system is called a vocoder). Unlike the acoustic model, these two parts of the framework may be entirely engineered and make use of no statistical methods. While modern deep TTS models are usually not referred to as SPSS, the SPSS pipeline as depicted in figure 1 applies just as well to those newer methods.

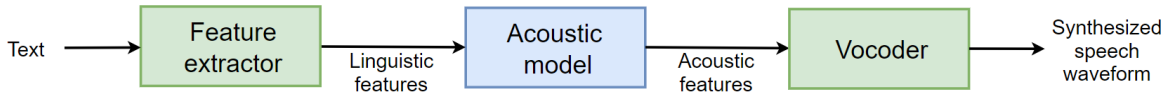

Figure 1: The general SPSS pipeline. The blue box is purely a statistical model while the green boxes can be engineered processes or/and statistical models.

The role of the feature extractor is to provide data that is more indicative of what the speech produced by the model is expected to sound like. Speech is a complex process, and directly feeding characters to a weak acoustic model will prove not to be effective. Providing additional features from natural language processing (NLP) techniques may greatly reduce the extent of the task to be learned by the acoustic model. It may however result in trade-offs when it comes to naturalness, especially for rare or unknown words. Indeed, manually engineered heuristics do not quite fully characterize all intricacies of spoken language. For this reason, feature extraction can also be done with trained models. The line between the feature extractor and the

[1] `https://google.github.io/tacotron/publications/speaker_adaptation/index.html`
[2] repo link

acoustic model can then become blurry, especially for deep models. In fact, a tendency that is common across all areas where deep models have overtaken traditional machine learning techniques is for feature extraction to consist of less heuristics, as models become able to operate at higher levels of abstraction.

A common feature extraction technique is to build frames that will integrate surrounding context in a hierarchical fashion. For example, a frame at the syllable level could include the word that comprises it, its position in the word, the neighbouring syllables, the phonemes that make up the syllable, ... The lexical stress and accent of individual syllables can be predicted by a statistical model such as a decision tree. To encode prosody, a set of rules such as ToBI (Beckman and Elam, 1997) can be used. Ultimately, there remains a work of feature engineering to present a frame as a numerical object to the model, e.g. categorical features are typically encoded using a one-hot representation.

One could wonder why the acoustic model does not directly predict an audio waveform. Audio happens to be difficult to model: it is a particularly dense domain and audio signals are typically highly nonlinear. A representation that brings out features in a more tractable manner is the time-frequency domain. Spectrograms are much less dense than their waveform counterpart and also have the benefit of being two-dimensional, thus allowing models to better leverage spatial connectivity. Unfortunately, a spectrogram is a lossy representation of the waveform that discards the phase. There is no unique inverse transformation function, and deriving one that produces natural-sounding results is not trivial. When referring to speech, this generative function is called a vocoder. The choice of the vocoder is an important factor in determining the quality of the generated audio.

Talk about evaluation metrics (mainly MOS)?

3 Evolution of the state of the art

3.1 In text-to-speech methods

The state of the art in SPSS has for long remained a hidden Markov models (HMM) based framework. (Tokuda, 2013). This approach, laid out in figure 2, consists in clustering linguistic features with a decision tree and to train a HMM per cluster (Yoshimura et al., 1999). The HMMs are tasked to produce a distribution over

The HMMs are trained to produce a distribution over mel-frequency cepstral coefficients (MFCC) with energy (called static features), their delta and delta-delta coefficients (called dynamic features) as well as a binary flag that indicates which parts of the audio should contain voice. This is shown in figure 3. A new sequence of static features is retrieved from these static and dynamic features using the maximum likelihood parameter generation (MLPG) algorithm (Tokuda et al., 2000). These static

features are then fed through the MLSA vocoder(Imai, 1983). It is possible to modify the voice generated by conditioning on a speaker or tuning the generated speech parameters with adaptation or interpolation techniques (Yoshimura et al., 1997) elaborate a bit on these techniques?, making HMM-based speech synthesis a multispeaker TTS system.

-¿ note that still inferior to concat

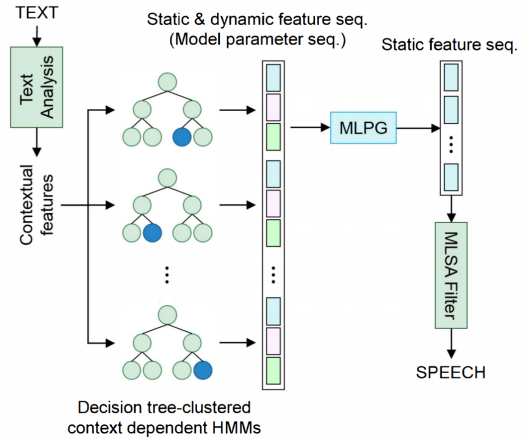

Figure 2: The general HMM-based TTS synthesis approach.

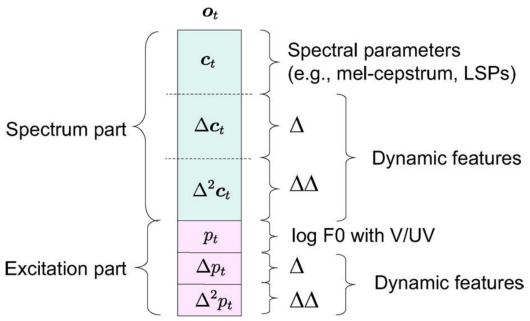

Figure 3: Dynamic and static features produced by the HMMs. F0 is the fundamental frequency and V/UV is the voicing flag.

Improvements to this framework were later brought by feed-forward deep neural networks (DNN), as a result of progress in both hardware and software. (Zen et al., 2013) proposes to replace entirely the decision tree-clustered HMMs in favor of a DNN. They argue for better data efficiency as the training set is no longer fragmented in different clusters of contexts, and for a more powerful model?. They demonstrate improvements over the speech quality with a number of parameters similar to that of the HMM-based approach. Their best model is a DNN with 4 layers of 256 units using a sigmoid activation function. Subjects assessing the quality of the generated audio samples report that the DNN-based models produces speech that sounds less muffled than that of the HMM-based models. Later researches corroborate these findings (Qian et al., 2014). (Hashimoto et al., 2015) additionally studies the effect of replacing MLPG with another DNN. The combinations of HMM/DNN and MLPG/DNN give rise to four possible frameworks, the novel ones being HMM+DNN and DNN+DNN[3], while HMM+MLPG and DNN+MLPG are the frameworks described respectively in (Tokuda, 2013) and (Zen et al., 2013). Each DNN they use is 3 layers deep with 1024 units using a sigmoid activation function. MOS results confirm that DNN+MLPG is significantly better than HMM+MLPG. The DNN+DNN approach performs as well as HMM+MLPG while HMM+DNN is worse. In another experiment, they introduce a DNN before and after MLPG. While both approaches yield a MOS similar to DNN+MLPG, neither are statistically significantly better. make this part nicer to read or maybe just remove it? Quid of the MOS? Mention DMDN?

[3]Note that since the two networks are consecutive in the framework, they can be considered a single network.

(Fan et al., 2014) supports that RNNs make natural acoustic models as they are able to learn a compact representation of complex and long-span functions, better so than shallow architectures such as the decision trees used in HMM-based TTS. Furthermore, the internal state of RNNs makes the mapping from an input frame to output acoustic features no longer deterministic, allowing for more variety in the synthesized audio. As RNNs are fit to generate temporally consistent series, the static features can directly be determined by the acoustic model, alleviating the need for dynamic features and MLPG. The authors present two RNNs: Hybrid_A with three feed-forward layers followed by one bidirectional (BDLSTM) layer and Hybrid_B with two feed-forward layers followed by two BDLSTM layers. They argue that deeper structures of BDLSTM would worsen the performance due to imprecise gradient computation. They compare these networks against the HMM and DNN based approaches described previously, using objective and subjective measures. The objective measures are compared with the ground truth: log spectral distance (LSD), voiced/unvoiced (V/UV) error rate and fundamental frequency (F0) distortion in root mean squared error (RMSE). The subjective measure is a preference test where participants must choose between two audio samples of different models and have the option to select neither. Results are shown in figures 4 and 5. DNN_A is 6 layers deep with 512 units per layer while DNN_B is 3 layers deep with 1024 units per layer. These two networks perform very similarly. Hybrid_B systematically performs better than the other approaches.

Measures / Model	LSD (dB)	V/U Error rate	F0 RMSE (Hz)
HMM (2.89M)	3.74	5.8%	17.7
DNN_A (1. 55M)	3.73	5.8%	15.8
DNN_B (2.59 M)	3.73	5.9%	15.9
Hybrid_A (2.30M)	3.61	5.7%	16.4
Hybrid_B (3.61M)	3.54	5.6%	15.8

Figure 4: Performance of the different frameworks evaluated on objective measures. In parentheses are the number of parameters of each acoustic model.

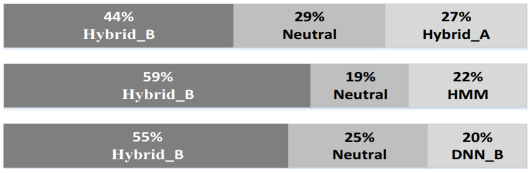

Figure 5: Two by two comparisons of some of the frameworks in terms of preferences.

Later, a substantial breakthrough in TTS is achieved with the coming of WaveNet (van den Oord et al., 2016). WaveNet is a deep convolutional neural network that, for a raw audio waveform, models the distribution of a single sample given all previous ones. It is thus possible to generate audio by predicting samples one at a time in an autoregressive fashion. WaveNet leverages stacks of one-dimensional dilated convolutions with a dilation factor increasing exponentially with the layer depth, which allows for a very large receptive field. The output is a categorical distribution (using a softmax layer) over sample values. For tractability reasons, the output signal is restricted to 256 values that correspond to a μ-law quantization, which is invertible. The MOS

resulting from one of the experiments suggests that there is no statistically significant degradation in the naturalness of speech quantized in this manner. The architecture of WaveNet comes with several non-trivial components, described in later sections of this document link to the section. On its own, a trained WaveNet generates sound alike the training data but without semantics. The network must be locally conditioned on linguistic contexts to achieve TTS synthesis. Furthermore, it allows for conditioning with respect to a vector constant at every timestep (global conditioning), which can be used to designate the speaker identity. The authors suggest that WaveNet is able to encode the embedding of several speakers seen in training with a shared internal representation. Speaker identities are given as a one-hot encoding. The audio generated by WaveNet requires no post-processing other than the inversion of the μ-law. TTS with WaveNet does not exactly fit as an SPSS system considering that it does not produce clear intermediate acoustic features and instead serves as both a statistical model and a vocoder in a single pipeline. Performance scores are reported in Figure 6. The parametric approach is an LSTM-based system while the other is an HMM-driven unit selection concatenative system (not detailed in this document). Notice how the results vary between US English and Mandarin Chinese, showing that TTS performance is not language agnostic. recheck all the info on WaveNet later + F0??

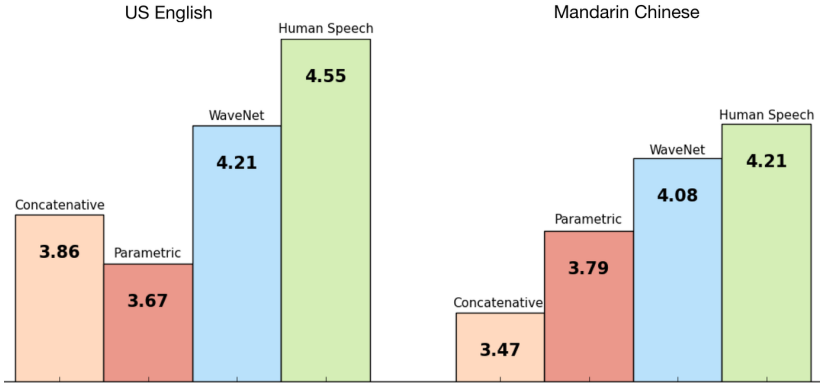

Figure 6: MOS of WaveNet's performance compared with a parametric and concatenative approach as well as with natural speech.

Deep Voice (Arik et al., 2017b) proposes a fully neural TTS framework that exploits WaveNet among other deep architectures. Deep Voice stands out by making use of only a few intermediate features: phonemes with stress annotations, phoneme durations, and F0. This has the advantage of making the framework easily transferable to new domains with little engineering effort as complex linguistic features need not be derived. The neural networks intervening in Deep Voice are: a grapheme-to-phoneme model that converts text to phonemes, a segmentation model that aligns a sequence of phonemes to an audio segment, a phoneme duration model that predicts the duration of each phoneme at generation time, a fundamental frequency model that predicts the V/UV flag with F0 values for voiced parts and finally, WaveNet which acts as an audio synthesis model. In all the works we presented previously, only the audio synthesis model was not manually engineered. While others have researched the use of neural networks for some of these components, Deep Voice is the first to simultaneously

employ them all in a single framework. However, the grapheme-to-phoneme model is only used as a fallback for words not present in a phoneme dictionary. The roles and interactions of these components at training and inference time are shown in figure 7. Deep Voice also improves on the inference time of WaveNet, with a speedup of up to 400. This allows for real-time or near real-time execution, with a tunable speed/quality trade-off. Deep Voice does not yield state of the art results but instead serves as groundwork for future researches. discuss results .

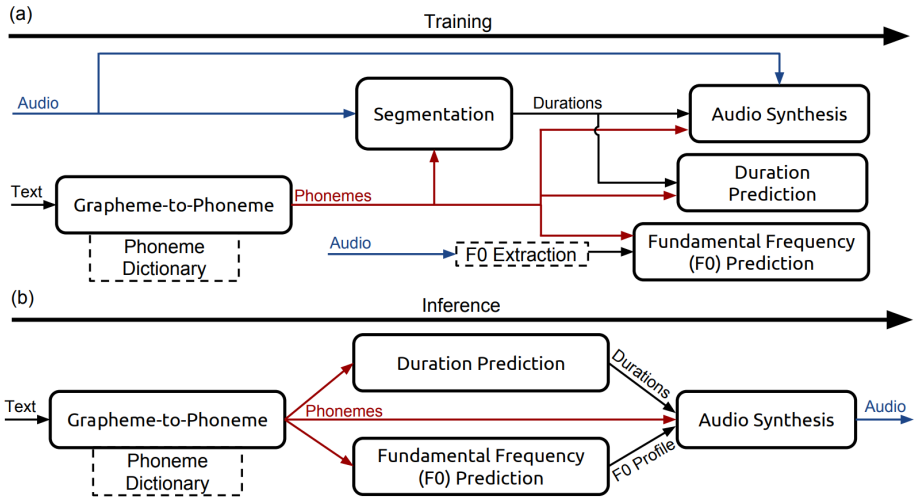

Figure 7: The training (a) and inference (b) procedure of Deep Voice. On the left of the image are the inputs, on the right are the outputs. Note that the segmentation model is only used for training.

Skipping the Deep Voice 2 & 3papers. I'll see later whether or not I should include them

Published approximately at the same time, Tacotron (Wang et al., 2017) is a sequence-to-sequence model that produces a spectrogram from a sequence of characters alone, further reducing the need for domain expertise. An audio waveform can be estimated from the spectrogram using the Griffin-Lin algorithm. Tacotron is also built fully with neural networks and is trained in an end-to-end fashion. It uses an encoder-decoder architecture where, at each step, the decoder operates on a weighted sum of the encoder outputs. This mechanism described in (Bahdanau et al., 2014) lets the network decide which steps of the input sequence are important with respect to each steps of the output sequence. Tacotron achieves a MOS of 3.85 on a US English dataset, which is more than the 3.69 score obtained in the parametric approach of (Zen et al., 2016) but less than the 4.09 score obtained by the concatenative approach of (Gonzalvo et al., 2016). The authors mention that Tacotron is merely a step towards a better framework. By the end of 2017, Tacotron 2 is published (Shen et al., 2017). The architecture of Tacotron 2 remains that of an encoder-decoder with attention although several changes to the types of layers are made. The main difference with Tacotron is the addition of a modified WaveNet for vocoder. On the same dataset, Tacotron 2 achieves a MOS of 4.53 compared to 4.58 for human speech (the difference is not

statistically significant), achieving the all-time highest MOS. In a preference study, Tacotron 2 was found to be only slightly less preferred on average than ground truth samples. The ratings from that study are shown in figure

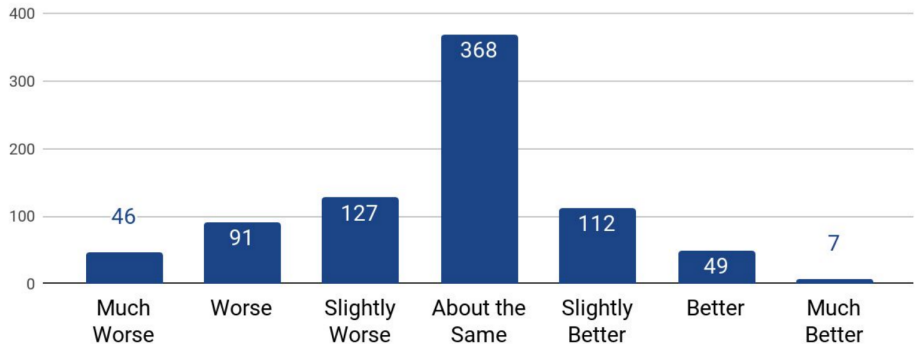

Figure 8: Preference ratings between Tacotron 2 and ground truth samples. 800 ratings on 100 items. The labels are expressed with respect to Tacotron 2.

Few samples

SV2TTS

Extensions?

Link audio samples for everything where available

4 Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis

To achieve our goal of real-time voice cloning, we reproduce (Jia et al., 2018) (referred as SV2TTS throughout this document). It describes an approach to multispeaker speech synthesis. A multispeaker speech synthesis framework is able to generate speech from text using different voices. This set of voices may be encoded into the model or be external to the model. In the first case,

This should come earlier

The approach used in SV2TTS is to build three models that are trained separately:

- A speaker encoder that derives an embedding from the short utterance of a single speaker. The embedding is a meaningful representation of the voice of the speaker, such that similar voices are close in latent space. This model is described in (Wan et al., 2017) (referred as GE2E throughout this document) and (Heigold et al., 2015).

Figure 9: Model overview. Each of the three components are trained independently.

- A synthesizer that, conditioned on the embedding of a speaker, generates speech features from a given text. This model is the popular Tacotron 2 (Shen et al., 2017) without WaveNet. The generated features cannot trivially be converted to an audio waveform.
- A vocoder that infers an audio waveform from the intermediate features generated by the synthesizer. The authors used WaveNet (van den Oord et al., 2016) as vocoder, effectively reusing the entire Tacotron 2 framework.

In the future, we will refer to these three models as respectively the encoder, the synthesizer and the vocoder. At inference time, the encoder is fed a reference utterance of the speaker to clone. It generates an embedding that is fed to the synthesizer, along with a selected text to generate. Finally, the vocoder takes the output of the synthesizer to generate the speech waveform.

Note that all models are interchangeable provided that they perform the same task. In particular, vanilla WaveNet is extremely slow for inference. Several later papers brought improvements on that aspect to bring the generation near real-time or faster than real-time, e.g. (van den Oord et al., 2017), (Paine et al., 2016). Note that in this context, real-time is achieved when the generation time is shorter than or equal to the duration of the generated audio. In our implementation, the vocoder used is based on WaveRNN (Kalchbrenner et al., 2018b).

- audio samples
encoder (or any):- a word about digital audio
encoder: - meaning of the paper title
synth: - the need of intermediate audio features
synth: - advantages of conditioning on embedding rather than

References

Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech, 2017a.

Sercan O. Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples, 2018.

Sercan Ömer Arik, Mike Chrzanowski, Adam Coates, Greg Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. Deep voice: Real-time neural text-to-speech. *CoRR*, abs/1702.07825, 2017b. URL `http://arxiv.org/abs/1702.07825`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL `http://arxiv.org/abs/1409.0473`.

Mary E. Beckman and Gayle Ayers Elam. Guidelines for tobi labelling, 03 1997.

Y Fan, Yuang Qian, Feng-Long Xie, and Frank Soong. Tts synthesis with bidirectional lstm based recurrent neural networks. pages 1964–1968, 01 2014.

Xavi Gonzalvo, Siamak Tazari, Chun-an Chan, Markus Becker, Alexander Gutkin, and Hanna Silen. Recent advances in google real-time hmm-driven unit selection synthesizer. In *Interspeech*, pages 2238–2242, 2016.

K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda. The effect of neural networks in statistical parametric speech synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4455–4459, April 2015. doi: 10.1109/ICASSP.2015.7178813.

Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. *CoRR*, abs/1509.08062, 2015. URL `http://arxiv.org/abs/1509.08062`.

S. Imai. Cepstral analysis synthesis on the mel frequency scale. In *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 93–96, April 1983. doi: 10.1109/ICASSP.1983.1172250.

Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez-Moreno, and Yonghui Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *CoRR*, abs/1806.04558, 2018. URL `http://arxiv.org/abs/1806.04558`.

Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aäron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. *CoRR*, abs/1802.08435, 2018a. URL `http://arxiv.org/abs/1802.08435`.

Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis, 2018b.

Tom Le Paine, Pooya Khorrami, Shiyu Chang, Yang Zhang, Prajit Ramachandran, Mark A. Hasegawa-Johnson, and Thomas S. Huang. Fast wavenet generation algorithm, 2016.

Y. Qian, Y. Fan, W. Hu, and F. K. Soong. On the training aspects of deep neural network (dnn) for parametric tts synthesis. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3829–3833, May 2014. doi: 10.1109/ICASSP.2014.6854318.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. *CoRR*, abs/1712.05884, 2017. URL `http://arxiv.org/abs/1712.05884`.

S. Shirali-Shahreza and G. Penn. Mos naturalness and the quest for human-like speech. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 346–352, Dec 2018. doi: 10.1109/SLT.2018.8639599.

K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for hmm-based speech synthesis. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 3, pages 1315–1318 vol.3, June 2000. doi: 10.1109/ICASSP.2000.861820.

Yoshihiko; Toda Tomoki; Zen Heiga; Yamagishi Junichi; Oura Keiichiro Tokuda, Keiichi; Nankaku. Speech synthesis based on hidden markov models. *Proceedings of the IEEE*, 101, 05 2013. doi: 10.1109/JPROC.2013.2251852.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016. URL `http://arxiv.org/abs/1609.03499`.

Aäron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. Parallel wavenet: Fast high-fidelity speech synthesis. *CoRR*, abs/1711.10433, 2017. URL `http://arxiv.org/abs/1711.10433`.

Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification, 2017.

Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: A fully end-to-end text-to-speech synthesis model. *CoRR*, abs/1703.10135, 2017. URL `http://arxiv.org/abs/1703.10135`.

Takayoshi Yoshimura, Takashi Masuko, Keiichi Tokuda, Takao Kobayashi, and Tadashi Kitamura. Speaker interpolation in hmm-based speech synthesis system. In *EUROSPEECH*, 1997.

Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In *EUROSPEECH*, 1999.

H. Zen, A. Senior, and M. Schuster. Statistical parametric speech synthesis using deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7962–7966, May 2013. doi: 10.1109/ICASSP.2013.6639215.

Heiga Zen, Yannis Agiomyrgiannakis, Niels Egberts, Fergus Henderson, and Przemyslaw Szczepaniak. Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices. *CoRR*, abs/1606.06061, 2016. URL `http://arxiv.org/abs/1606.06061`.