Text-To-Speech Synthesizer and Voice Cloning using Generative Model

Ravishek Kumar Singh
Sharda University,
Greater Noida,UP(India)

Himanshu Pal
Sharda University,
Greater Noida,UP(India)

Rohit Raj
Sharda University,
Greater Noida,UP(India)

Mohd Tariq
Sharda University,
Greater Noida,UP(India)

Sandeep Kumar
Sharda University,
Greater Noida,UP(India)

Abstract--**We present a neural network-based text-to-speech (TTS) synthesis system that can synthesise spoken sounds in the voices of many speakers. Our system is made up of three independently trained components: a speaker encoder network that was trained on a speaker verification task using an independent dataset of noisy speech without transcripts from thousands of speakers to generate a fixed-dimensional embedding vector from only seconds of reference speech from a target speaker; a Tacotron-based sequence-to-sequence synthesis network that generates a model spectrogram from text, conditioned on the speaker embedding; and we show that the proposed model can transfer the discriminatively-trained speaker encoder' s knowledge of speaker variability to the multispeaker TTS challenge and synthesis authentic speech from speakers not observed during training. To get the optimum generalisation performance, we quantify the value of training the speaker encoder on a wide and varied speaker set. Finally, we demonstrate that randomly chosen speaker embeddings can synthesis speech in the voices of fresh speakers who are not comparable to those used in training, showing that the model has learnt a high-quality speaker representation.**

Keywords—Tacotron, spectrogram, TTS, embeddings.

INTRODUCTION

Our objective is to create a system that can generate natural speech for the speaker in the most efficient way possible. We employ a zero-shot learning setup in which we use a few seconds of reference audio to synthesise a new speech in the speaker's voice. Accessibility applications for such systems include regaining the capacity to speak normally to users who have lost their voice and are thus unable to give many fresh training instances. They may also open the door to new applications, such as transferring a voice between languages for more authentic speech-to-speech translation or creating realistic speech from text in low-resource circumstances. However, it is also vital to consider the possibility of misusing this technology, such as imitating someone's voice without their permission. To address safety issues in accordance with concepts such as [1,] we demonstrate that sounds generated by the proposed model may be clearly differentiated from actual voices. Natural speech synthesis necessitates training on a large number of high-quality speech-transcript pairings, and supporting many speakers often necessitates tens of minutes of training data per speaker [8]. It is impossible to record a huge volume of high-quality data for a big number of speakers. Our method separates speaker modelling from speech synthesis by building a speaker-discriminative embedding network that captures the space of speaker characteristics separately from constructing a high-quality TTS.

model using a smaller dataset conditioned by the first network's representation By decoupling the networks, they may be trained on separate data, reducing the demand for high-quality multispeaker training data. On a speaker verification challenge, we train the speaker embedding network to detect whether two different utterances were delivered by the same speaker. Unlike the succeeding TTS model, this network is trained on untranscribed speech from a large number of speakers with reverberation and background noise.

We show that speaker encoder and synthesis networks may be trained on imbalanced and disconnected speaker sets while still generalising successfully. We train the synthesis network on 1.2K speakers and demonstrate that training the encoder on a considerably larger collection of 18K speakers enhances adaption quality and allows us to synthesise fully fresh speakers by sampling from the embedding prior.

Final training of TTS models, which are trained directly from text-audio pairings without the need of hand- crafted intermediate representations, has received a significant amount of attention [17, 23]. Tacotron 2 [15] employed WaveNet [19] as a vocoder to reverse spectrograms created by an encoder-decoder architecture with attention [3,] achieving naturalness similar to human speech by combining Tacotron's [23] prosody with WaveNet's audio quality. It could only accept one speaker. Gibiansky et al. [8] presented a multispeaker Tacotron variant that learnt low-dimensional speaker embedding for each training speaker. Deep Voice 3 [13] presented a fully convolutional encoder-decoder architecture that could accommodate over 2,400 LibriSpeech [12] speakers. These systems only handle the synthesis of voices seen during training since they learn a predetermined set of speaker embeddings. VoiceLoop [18], on the other hand, offered a unique architecture based on a fixed-size memory buffer that may create speech from unheard sounds during training. For a new speaker, obtaining decent results requires tens of minutes of enrollment speech and transcripts.

Recent enhancements have enabled few-shot speaker adaptation, in which only a few seconds of speech (without transcripts) from each speaker may be utilised to produce new speech in that speaker's voice. [2] extends Deep Voice 3 by comparing a speaker adaptation method similar to [18], in which model parameters (including speaker embedding) are fine-tuned on a small amount of adaptation data, to a speaker encoding method, in which a neural network predicts speaker embedding directly from a spectrogram. The latter method is far more data economical, achieving increased naturalness with little quantities of adaptation data in as few as one or two utterances. It is also much more computationally efficient because it does not need hundreds of backpropagation iterations. Similarly, Nachmani et al. [10] enhanced VoiceLoop to anticipate a speaker embedding using a target speaker encoding network. This network is trained alongside the synthesis network using a contrastive triplet loss to guarantee that embeddings predicted from the same speaker are more similar than embeddings computed from different speakers. A cycle-consistency loss is also applied to ensure that the synthesised speech encodes to the same embedding as the adaptive utterance. A comparable spectrogram encoder network was demonstrated to function for transferring target prosody to synthetic speech [16]. We show that training a comparable encoder to differentiate between speakers results in a reliable transfer of speech characteristics. Our approach is most comparable to the speaker encoding models in [2, 10], with the exception that we use an independently trained network for speaker verification.

task employing a state-of-the-art generalised end-to-end loss on a huge dataset of untranscribed audio from tens of thousands of voices [22]. [10] used a similar speaker-discriminative representation in their model, but all components were trained together. We investigate transfer learning from a pre-trained speaker verification model in comparison.

Doddipatla et al. [7] employed a similar transfer learning setup to condition a TTS system, using a speaker embedding calculated from a pre-trained speaker classifier. We use an end-to-end synthesis network that does not rely on intermediate language characteristics and a significantly different speaker embedding network that is not constrained to a closed set of speakers in this article. Furthermore, we investigate how quality varies with the number of speakers in the training set and discover that zero-shot transfer necessitates training on thousands of speakers, far more than the amount employed in [7]. speaker similarity assessment.

1. Literature survey

Our method for real-time voice cloning is heavily influenced by (Jia et al., 2018). (referred to as SV2TTS throughout this document). It proposes a framework for zero-shot voice cloning with just 5 seconds of reference speech required. This paper is only one of numerous Tacotron series5 articles written at Google. Interestingly, the SV2TTS study does not provide any new ideas; rather, it is built on three key prior Google works: the GE2E loss (Wan et al., 2017), Tacotron (Wanget al., 2017), and WaveNet (van den Oord et al., 2016). The entire architecture is a three-stage pipeline, with the stages corresponding to the models given in the preceding sequence. Many of Google's existing TTS tools and functions, such as the Google assistant6 and Google cloud services[7], rely on these similar models. While there are several open-source reimplementations of these models available online, none of the SV2TTS frameworks are (as of May 2019).

A. Voice Cloning.

In various papers [8], [9], the phrase voice cloning refers to a specific speaker adaption scenario for TTS with untranscribed speech. In this research, we use the term "voice cloning" to refer to any form of system that makes speech by replicating the voice of a specific speaker. The fundamental distinction between voice cloning and speech synthesis is that the former emphasises the target speaker's identity [10], whilst the latter occasionally disregards this feature for naturalness.

Voice conversion is a closely related task to voice cloning. The purpose of voice conversion is to alter the sound of an utterance from the source speaker to that of the target speaker while keeping the linguistic contents untouched. Voice conversion techniques, unlike voice cloning, do not need to generalise to unseen texts. To synchronise the spectra of separate speakers, one typical way is dynamic frequency warping. Roupakia and Agiomyrgiannakis [2016]

B. Training voice cloning model.

The traditional VC technique is text-dependent, requiring concurrent utterances of source and target speakers as training data [10], [12]. Because acquiring these utterances is an expensive and time-consuming task, a parallel VC system is sometimes created with as little as five minutes of data from a speaker [19]. This is cumbersome, and it lowers the overall quality of VC systems. Many people have worked on methods for developing VC systems with non-parallel utterances [5]. We may use HMM models to create a transformation function to adjust pretrained models to non-parallel speech [18].

C. Text-to-Speech.

Typically, a TTS system is trained on many of hours of transcribed speech [9], [17]. Because of the great need for quantity and quality, a professional voice feeder is frequently hired to capture such data in a controlled setting. As a result, the traditional technique is unsuitable for voice cloning tasks in which we have no control over the target speaker, recording environment, or volume of data. We can modify a pretrained model to develop a TTS system for speakers with a minimal amount of labelled data.

The first model may be trained using data from a single speaker [11] or data from numerous speakers [18]. When the data from target speakers is adequate (e.g., one hour), this easy fine-tuning provides a high-quality model [7]. When data is extremely restricted (e.g., one minute), we might confine tuning to certain components rather than the entire network to avoid overfitting [13]. In essence, speaker adaptation transfers information obtained from large amounts of data from one or more speakers in order to lessen demand on a target.

2. METHODOLOGY

Our voice cloning model is text to speech model (TTS). It is based on Speaker encodings and the way speech is styled. In this a term Global Style Tokens comes in frame that is TTS models learns from the library of latent style vector called GST.As we find that in using a addition of good style information in TTS model provides control over style of synthesized speech and also produces more natural and realistic sound for the new speaker for which it is not trained .The main terms that is used in this TTS Speaker encoder , Speaker adaptation, Speaker Classification, Speaker Verification MEL-spectrogram Synthesizer, Cloning techniques, Zero-shot and Model adaptation. Speaker Encoder. Speaker adaptation after conversion into waveform is done and then speaker classification then speaker verification. we need few samples to generate the speech like after training the model with few samples it can generate speech this is called Few-shot Generative Modelling. This speech generator model is trained on different speaker of different assent say the same sentence in different ways hence different spectrograms mapped to the same text, now this model is able to generate less accurate model, but when we provide the detail of speaker like gender, age etc then the model understands the difference and able to generate more accurate model. The three main components Speaker Encoder, Mel Spectrogram Synthesizer and Vocoder are all trained and described below in detail which are used to synthesize the voice of unknown speaker from few known speakers.

2.1. Speaker Encoding

In the TTS model, speaker encoding is done using speaker embedding, which is started at random and trained end-to-end with a synthesizer. Speaker embedding is a simple method to show speaker's identity in a compressed way Speaker encoding is a process to of speaker embedding from audio training speech data which is of unknown

speaker. Although English is the most widely spoken language, the accents of speakers vary. Some speakers can speak fluently, while others can speak at the same pitch. As our sample size grows, we find a wide range of accents. The speaker adaptation process should be used here because when training and test data are mismatched, the model's performance gradually declines.

Speaker embedding searches the available speakers for speakers that are not in the training dataset. To achieve the goal of voice cloning in TTS models to adapt fromvarious speakers.

It is a method of compressing a speech signal and then converting it to binary digits. Every time We feed the model or train the model with new training speech, and the encoder compresses the speech signal before converting it to binary digits. Speech compression is used to store speech and to encrypt messages. By averaging the embeddings of each individual segment, the final embedding is calculated.

2.2. Mel-Spectrogram Synthesizer

It is a method to get good features of audio and visualize them into pictures. This is a flavour of spectrogram that is extensively used in ai audio research and real world application. In y axis we have time and in x-axis we have a frequency and each point in between shows as how the frequency is at every point of time. Now frequency representation with a normal vanilla spectrogram is linear and it uses hunts and this is kind of problematic of the way we perceive frequency .If we have couple of audio samples so each pair have pair of notes and if has same one more then when we look up at the frequency mapping or can say the way the frequency express the main two difference we can notice for both is same around 200 Hz. Now different frequency in signal evolve over time we want this feature to have perceptually relevant amplitude evolve overtime and invisible to feature to have personal relevant amplitude representation and this once again is because amplitude and the way we perceive it is logarithmic it's not linear right and both of these things we can achieved with vanilla spectrograms and indeed with have a logarithmic amplitude spectrograms the but we cannot achieve with vanilla spectrogram is the third expect which is perceptually relevant frequency and this is what male expect programs are all about and write they check out all of this. But now male spectrogram so it's a term it's a conceptwith two words one of it is spectrogram and second is mel.

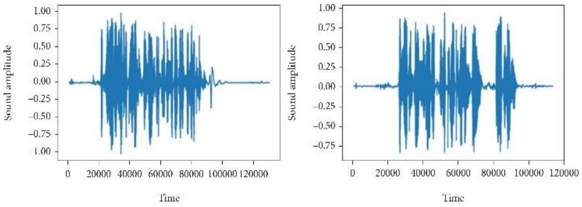
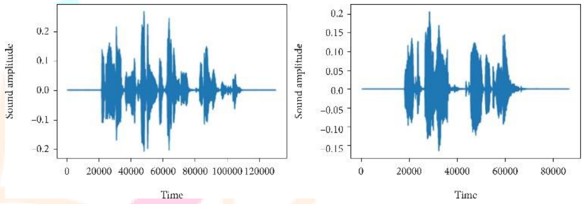


Fig: Waveform of the female voice.
Fig: Waveform of the male voice.

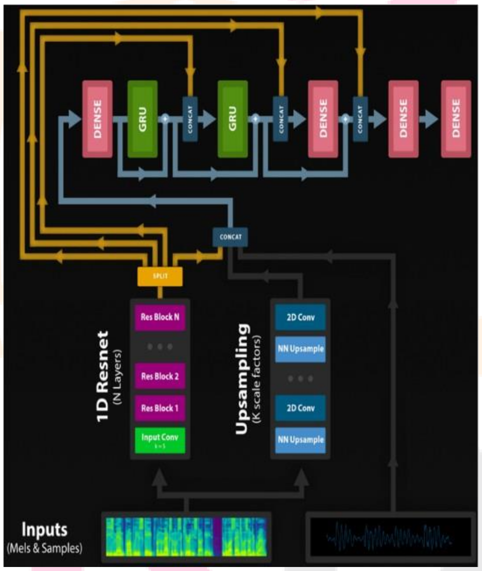


The recipe to extract MEL- spectrogram first is extract STFT second is to convert to DBS and third is to convert the frequency to mel. Now the two steps are like the once we use for Vanilla spectrogram so nothing new here the new thing is that third step here which is converting the frequencies to mel representation. These are the three steps to convert the frequencies to mel scale. First choose number of mel bands second is to construct mel filter banks and third is apply mail filter banks to spectrum.

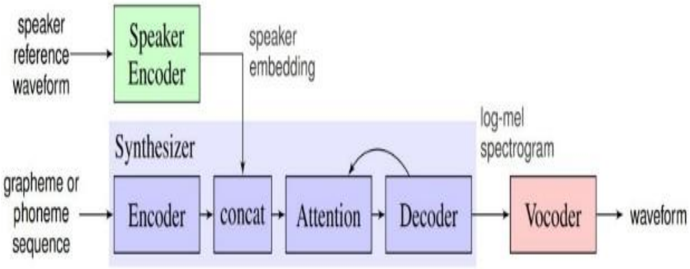
2.3. Cloning Techniques: Zero-Shot and Model Adaptation

Fig: The alternative RNN Architecture.

In zero-shot voice cloning, we explain the speaker embedding by taking the mean and then L-2 normalising the speaker encodings of the target speaker's individual samples. Because speaker encodings are obtained directly from waveforms, zero-shot voice cloning does not require audio transcriptions of the new speaker. Model Adaptation: When transcribed samples of a new speaker are available, we can fine-tune our synthesis model using the text and audio pairs. Because the whole model adaptation allows for more degrees of freedom, fine-tuning the entire synthesis model is faster and more effective than fine-tuning only the speaker embedding layer. Our preliminary model adaptation experiments indicated the same conclusion. We hypothesise that fine-tuning the synthesis model's final few layers is required, if not sufficient, for adapting the synthesiser to speaker-specific speech characteristics. As a result, we investigate the two model adaptation techniques listed below: Overall adaptation - Fine-tune all synthesis model parameters on the new speaker's text and audio pairs. Adaptation decoder - Fine-tune only the synthesis model's decoder parameters. The benefit of only modifying the decoder parameters is that it requires fewer speaker-specific model parameters, and in a real-world deployment setting, a shared encoder can be used across all speakers.



2.4. Speaker Adaptation



The goal of speaker adaptation is to fine-tune a trained multi-speaker model for an unknown speaker using a few audio-text pairs. Either the speaker embedding or the speaker itself can be fine-tuned. or even the entire model. Only for embedding customization. Although the entire model provides more degrees of freedom for speaker adaptation, it is difficult to optimise for a small amount of cloning data. Early stopping is required to avoid overfitting. Assume we have a high-recognition-accuracy acoustic model for one speaker and want to know how to improve it for another speaker using only a few utterances of his/her speech data. It is composed of thousands of states, each with a Gaussian mixture distribution and a variety of mixture components. A covariance matrix and a mean vector with dozens of elements are used to represent each component of the mixture. It also includes parameters for transition and initial probability. Each utterance lasts approximately one second, with a feature vector containing dozens of elements available every 10 milliseconds. This means that from a few utterances, only a few thousand data samples can be obtained. In this case, ML estimation using the EM algorithm is unable to precisely estimate the model parameters. As a result, recognition accuracy would be significantly lower than before. This is known as the data sparseness issue. Speaker adaptation is defined as a process that employs adaptation data to determine a mapping function f from the initial model's parameter space to the target model's parameter space. Recognition accuracy should improve with even a small amount of adaptation data. As the amount of adaptation data increases, recognition accuracy should asymptotically approach that of a matched model.

Fig: The TSS framework during inference. The blue blocks represents the high-level view of the Tacotron architecture modified to allow conditioning on a voice.

An audio sample's speaker classifier determines which speaker it belongs to. A speaker classifier is trained with the set of speakers used for cloning to evaluate voice cloning. High classification accuracy would result from high-quality voice cloning the architecture is made up of spectral and temporal processing layers that are similar to each other, as well as an additional embedding layer before the softmax function. Speaker verification is the process of authenticating a speaker's claimed identity based on a test audio and enrolled audios from the speaker. It uses binary classification to determine whether the test audio and enrolled audios are from the same speaker. We consider a text-independent speaker verification model from start to finish the speaker verification model can be trained on a multi-speaker dataset and then used to determine whether the cloned and ground-truth audio are from the same speaker. Unlike the speaker classification approach, the speaker verification model does not require cloning training with audios from the target speaker, so it can be used for unseen speakers with a few samples . The equal error- rate, as a quantitative performance metric, can be used to determine how close the cloned audios are to the original audios.

3. Conclusion

We investigate two neural voice cloning approaches: speaker adaptation and speaker encoding. We show that even with a small number of cloning audios, both approaches can achieve good cloning quality. We demonstrate that speaker adaptation and speaker encoding can both achieve naturalness. A MOS resembling the baseline multi-speaker generative model as a result, the proposed techniques could be improved in the future with better multi-speaker models (for example, replacing GriffinLim with WaveNet vocoder). We show that both approaches benefit from a larger number of cloning audios in terms of similarity. The difference in performance between whole-model and embedding-only adaptation suggests that, in addition to speaker embeddings, some discriminative speaker information remains in the generative model. The advantage of compact representation via embeddings is fast cloning and a small footprint per speaker. We observe drawbacks of training the multi-speaker generative model with a speech recognition dataset with low-quality audios and limited speaker diversity. Increases in dataset quality would result in greater naturalness. We anticipate that a large-scale and high-quality multi-speaker dataset will significantly benefit our techniques.

References

1. *A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for rawaudio. arXiv preprint arXiv:1609.03499, 2016a.*

2. *A. v. d. Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In Advances in Neural Information ProcessingSystems, 2016b.*

A. v. d. Oord, S. Dieleman, H. Zen,
K. Simonyan, O. Vinyals, A. Graves,
N. Kalchbrenner,

3. *A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In NIPS. 2017*

4. *B. M. Lake, C. ying Lee, J. R. Glass, and J. B. Tenenbaum. One-shot learning of generative speech concepts. In CogSci, 2014.*

5. *B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level conceptlearning through probabilistic program induction. Science, 2015.*

6. *B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. One-shot learning byinverting a compositional causal process. In NIPS, 2013.*

7. *D. Rezende, Shakir, I. Danihelka, K. Gregor, and D. Wierstra. One-shot generalization in deep generative models. In ICML, 2016.*

8. *D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur. Deep neural network- based speaker embeddings for end- to-end speaker verification. In IEEE Spoken Language Technology Workshop (SLT), pages 165–170, 2016.*

9. *J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen,*
Y. Zhang, Y. Wang, R. SkerryRyan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. arXiv preprint arXiv:1712.05884, 2017.

10. *J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and*
Y. Bengio. Char2wav: End-to-end speech synthesis. 2017.

11. *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.*

12. *S. Prince and J. Elder. Probabilistic linear discriminant analysis for inferences about identity. In ICCV, 2007. S. E. Reed, Y. Chen, T. Paine,*

A. van den Oord, S. M. A. Eslami,
D. J. Rezende. CoRR, 2017.

13. *stability, and variation. CoRR, abs/ 1710.10196, 2017.*

14. *T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing ofgans for improved quality,*

15. *V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: anASR corpus based on public domain audio books. In IEEE ICASSP, 2015.*

16. *W. Ping, K. Peng, A. Gibiansky, S. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller. Deep Voice 3:Scaling text-to-speech with convolutional sequence learning. In ICLR, 2018.*

17. *X. Li and X. Wu. Modeling speaker variability using long short-term memory networks for speech recognition. In INTERSPEECH, 2015.*

18. *Y. Miao and F. Metze. On speaker adaptation of long short-term memory recurrent neural networks. In Sixteenth Annual Conference of the International Speech Communication Association, 2015.*

19. *Y. Miao, H. Zhang, and F. Metze. Speaker adaptive training of deep neural network acoustic modelsusing i-vectors. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015.Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani. Voiceloop: Voicefitting and synthesis via aphonological loop. In ICLR, 2018.*