Text-To-Speech Synthesis using Melspectograms and WaveNet

Devansh Dixit[1], Aparna Joshi[2], A. Dhanancheyan[3], Divanshu Tiwari[4], Gaurav Bhatt[5]

[1,3,4,5]B. E Students, Department of Information Technology, Army Institute of Technology, Pune

[2]Assistant Professor, Department of Information Technology, Army Institute of Technology, Pune

Abstract: *In traditional approaches for text-to-speech, a large amount of input text has to be provided. The proposed model uses neural approach for text to speech synthesis using melspectograms and waveNet. The model design consists of three main parts, Encoder, Synthesizer and Vocoder. The general idea is to extract the voice essence using a small audio sample. After that comes the synthesis of melspectrogram with the extracted voice essence and new speech content from text. Then finally, the melspectrogram is used to generate voice, which is similar to the input audio sample but with entered text content. The main feature of this model is that it uses much smaller audio input for sampling, hence lesser training time as compared to the conventional method. The applications of this model include artificial voice generation, video games, speech generation and many more.*

Keywords: **We would like to encourage you to list your keywords in this section**

1. Introduction

Generating speech from text is the task of training audio from a user and using it to generate vocals not present in the training data. In traditional approaches for text-to-speech, a large amount of input text has to be provided. But this can also be done by using just five seconds of input from user. This approach is known as zero-shot approach, where a few seconds of transcribed reference audio from a target speaker is used to synthesize new speech in that speaker's voice, without updating any model parameters. In-depth learning models are prominent in many areas of machine learning used. Text-to-speech (TTS), the process of integrating input speech from a text message, is not. In-depth models will produce a much more natural expression of nature than the corresponding traditional methods that first emerged in 2016. Well-recorded speech data sets are a small resource. Combining natural voice with proper pronunciation, healthy voice and minimal background sound requires training data with similar qualities. In addition, data efficiency remains a major problem for in-depth learning. Training a typical model to attend a talk like Tacotron requires talking for hundreds of hours. However, the ability to make a speech with any word is appealing to many types of apps,

be it practical or just a matter of customization. Research has led to the principles of voice modification and voice integration. While complete training of a single TTS speaker model is actually a method of voice integration. A common way to set up a TTS model is to install new speakers in voice embedding to be measured. Embedding is low in size and is based on the speaker encoding model that takes the reference expression as input. This method works with more data than training a different TTS model for each speaker, in addition to being orders for size faster and less expensive to calculate. Interestingly, there is a great deal of variation in the length of spoken speech that is needed to integrate the word between different ways, from half an hour for each speaker to just a few seconds. This feature usually determines the similarity of the voice made in relation to the actual speaker voice. Our goal is to achieve the strongest kind of voice integration. The emerging frame should be able to work with a zero-shot rotation, that is, of an invisible speaker during training. It should include a speaker's voice with a few seconds of reference speech. In addition, we are compiling a model based on framework [1] to make it work in real time, that is, to produce a speech in a shorter time or equal to the time a speech is produced.

2. Literature Survey

Statistical parametric speech synthesis (SPSS) refers to a group of Text-to-Speech (TTS) data-driven methods that emerged in the late 90's. In SPSS, the relationship between the features listed in the input text and the output. Acoustic features are studied in the mathematical model produced. The complete SPSS framework also includes a pipeline for the components to be integrated, as well as a system that can recreate the audio form from the acoustic features produced by the acoustic model. Unlike the acoustic model, these two parts of the frame may be completely fabricated and do not use mathematical methods [2]. The role of the feature output is to provide details that show exactly how the speech produced by the model is expected to sound. Speech is a complex process, and direct feeding of characters on a weak acoustic model would seem ineffective. Providing additional features from natural language processing techniques (NLP) can significantly reduce the magnitude of the work that can be learned by the acoustic model. For this reason, feature removal can also be performed with trained models. The state of the art in SPSS has for long remained a hidden Markov model (HMM) based framework. 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. [3] proposes to replace entirely the decision tree-clustered HMMs in favour of a DNN. Researchers argue for better data efficiency as the training set is no longer fragmented in different clusters of contexts. Researchers also demonstrate improvements over the speech quality with a number of parameters similar to that of the HMM-based approach. Researchers were support that RNNs make natural acoustic models as they are able to learn a compact representation of complex and long-span functions. 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. They compare networks of bidi- reactional LSTMs against the

HMM and DNN based approaches described previously. Follows Tacotron, a sequence-to-sequence model that produces a spectrogram from a sequence of characters alone, further reducing the need for domain expertise. In this framework, the vocoder is the Griffin-Lim algorithm. Tacotron uses an encoder-decoder architecture where, at each step, the decoder operates on a weighted sum of the encoder outputs [4]. Tacotron complete framework is a three-stage pipeline, where the stages of the framework are a speaker encoder, a synthesizer and vocoder [4]. Our approach to real-time voice cloning is largely based on zero-shot voice cloning that only requires 5 seconds of reference speech [5].

3. Proposed Architecture

This section describes the proposed architecture of system refer figure 1.

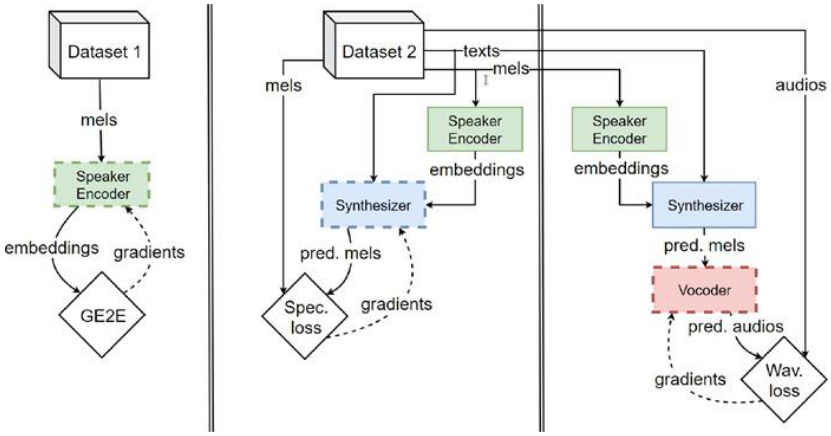

Figure 1: The sequential three-stage training

The proposed method has three main components, Encoder, Synthesizer and Vocoder. A common idea is to remove the vocal cords using a small volume sample. Next comes the integration of a melspectrogram with voice recognition and new speech content from the text. Finally, a melspectrogram is used to produce sound, similar to a sound sample but with embedded text content. An important feature of this model is that we use small audio inputs to make examples of that, which is why the minimum training time grows in a normal way. It requires training of all three models on the same database, this database may need to meet the requirements of all models: a large number of Encoder speakers but at the same time, synthesizer scripts. Low synthesizer sound and somehow normal sound quality encoder. These are conflicts they are problematic and can lead to training models that can do better if trained separately on different data sets. Specifically, a small database is possible led to poor construction and thus poor zero-shot performance. Integrated model integration can be very difficult to achieve. In particular, the Tacotron synthesizer can take a significant amount of time before producing the right alignment. A practical way to deal with the second issue is to separate the training of synthesizer and vocoder. If you think the encoder has been cheated, the synthesizer can train to directly predict melspectrograms of audio target. It always works well with the speaker encoder. Unlike synthesizer and vocoder, encoder has no labels to be trained. The work has been

described as producing" logical" embedding that produces a voice in speech. One might think of how to train a speaker encoder like an autoencoder, however may require that the corresponding sampling model be identified in the text prediction. Whether the database is forced into the same sentence, any need text and model for nomination by synthesizer. In both cases the quality of the training is disabled by the database and is unlikely to increase properly. Fortunately, the loss of GE2E brings a solution to this problem and allows for the training of the speaker encoder without a synthesizer. While all parts of the frame are trained separately, it is still necessary for the synthesizer to have embedded from a trained codec(encoder) and for the vocoder to have melspectrograms from a trained synthesizer. The speaker encoder needs to be optimized enough to produce a reasonable embedding in the synthesizer database; even if you are generally trained data, should still be able to work with zero setting during reflection.

4. Experimental Results

Two public datasets we used for training the synthesizer and vocoder networks. VCTK database contains more than 40 hours of speech from 109 speakers, the majority of which have British accents [2]. The database was split into three subsets: train, validation and test. LibriSpeech contains 436 hours of speech from 1,172 speakers, sampled at 16 kHz. Most of the speakers are United States English [6]. The speaker sets in this dataset have no same samples among the train, validation, and test sets [6]. Mean Opinion Score (MOS) evaluations based on subjective listening tests. The rating scores range from 1 to 5 and are in 0.5 points of increment. This framework is used to evaluate synthesized speech on two parameters: its naturalness and similarity to real speech from the target speaker [7].

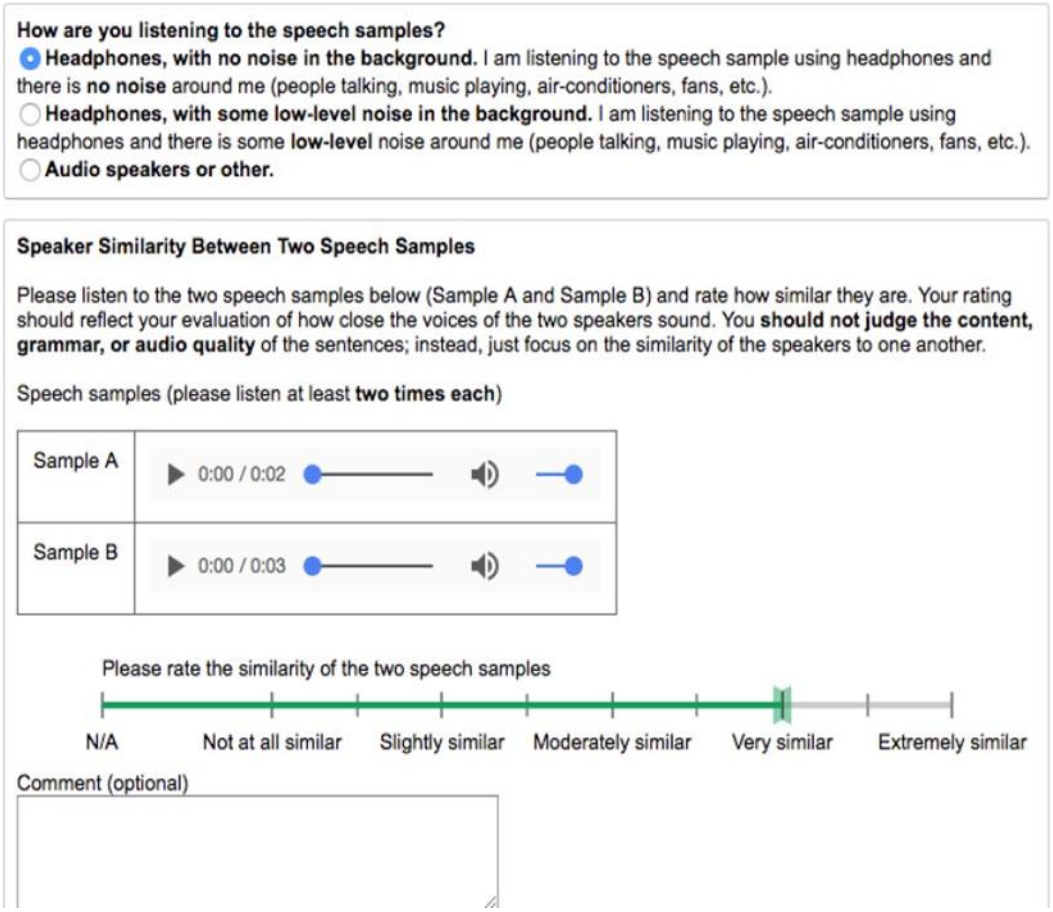

Figure 2: GUI of Mean Opinion Score evaluation for similarity

The naturalness of synthesized speech using synthesizers and vocoders trained on LibriSpeech and VCTK datasets are compared. An evaluation set of 100 phrases was constructed which are not present any training sets. Two sets of speakers were the evaluated for each model: one comprising of speakers in the train set (Seen), and another comprising of those that were not included (Unseen). 11 seen and unseen speakers were used for VCTK and 10 seen and unseen speakers for LibriSpeech. To check how well the synthesized speech gets close to that from the target speaker, we compared each synthesized speech instance with a randomly selected ground truth speech instance from the same speaker. Each pair is evaluated by one person. The evaluator is given the following instructions:" Just focus on the similarity of the speakers to one another and not judge the content, grammar, or audio quality of the sentences".

Table 1: Mean Opinion Score for Naturalness of Speech

System	VCTK Seen	VCTK Unseen	LibriSpeech Seen	LibriSpeeach Unseen
Ground truth	4.43 ± 0.05	4.49 ± 0.05	4.49 ± 0.05	4.42 ± 0.07
Proposed model	4.07 ± 0.06	4.20 ± 0.06	3.89 ± *0.06*	4.12 ± 0.05

Table 2: Mean opinion Score for Speaker Similarity

System	Speaker set	VCTK	LibriSpeech
Ground truth		4.67± 0.04	4.33 ± 0.08
Proposed model	Seen	4.22 ± 0.06	3.28 ± 0.08
Proposed model	Unseen	3.28 ± 0.07	3.03 ± 0.09

The PCA graph refer figure 3, helps us visualize the speaker embeddings extracted from LibriSpeech utterances. (left) shows that synthesized speech utterances are very close to real speech from the same speaker. However, synthetic utterances are still very easy to distinguish from the real human speech. This difference can be demonstrated by the t-SNE visualization (right) where synthetic and real speech from the same speaker form adjacent yet distinct clusters.

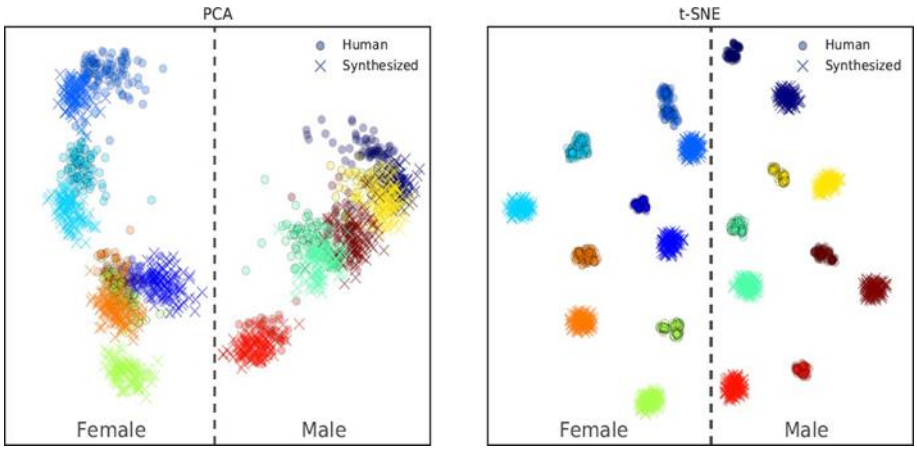

Figure 3: Speaker embeddings visualization from LibriSpeech utterances.

5. Conclusion

In the proposed model, neural approach is used to develop a model for zero shot, real-time voice cloning. The framework consists of three major components: Encoder, Synthesizer and Vocoder. Although the results can be considered quite close to the actual human speech, they are still clearly distinguishable from the real speech. Despite some unnatural prosody, the results can be considered satisfactory and the voice cloning ability of the framework can be considered reasonably good, although not as good as models that use more reference speech time.

References

[1] 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 synthe- sis, 2018.

[2] 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 multi- speaker text-to-speech synthesis, 2018

[3] Heiga Zen, Yannis Agiomyrgiannakis, Niels Egberts, Fergus Henderson, and Przemys- law Szczepaniak. Fast, compact, and high-quality LSTM- RNN based statistical parametric speech synthesizers for mobile devices, 2016.

[4] 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, 2017

[5] 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 multi- speaker text-to-speech synthesis, 2018

[6] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from lib- rispeech for text-to-speech , 2019

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

[8] 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 synthe- sis, 2018.

[9] Tom Le Paine, Pooya Khorrami, Shiyu Chang, Yang Zhang, Prajit Ra-machandran, Mark A. Hasegawa-Johnson, and Thomas S. Huang. Fast wavenet generation algo- rithm, 2016.

[10] 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, 2016.

[11] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression, 2017.

[12] Tom Le Paine, Pooya Khorrami, Shiyu Chang, Yang Zhang, Prajit Ra-machandran, Mark A. Hasegawa-Johnson, and Thomas S. Huang. Fast wavenet generation algo- rithm, 2