Team 18: **Transfer Learning from Speaker Identification to Zero-Shot Multispeaker Text-To-Speech Synthesis in Korean**

20150144 김상우, 20193138 김재윤, 20160171 김진우, 20193649 하진철

We chose <option 2>. Our git repo is here. We highlighted our novel modifications to the model.

Introduction

Current Korean text-to-speech (TTS) systems rely on minutes to hours of training voice data from target speaker, and therefore are not capable of synthesizing voice from limited data or of unseen ones. To address the problem, we aimed to build a TTS system that generates a Korean dialogue from text in the voice of unseen speaker, given only a few seconds of speech audio.

Methods

For the task, we applied the model in the following paper: **Ye Jia et al., 'Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis', *NIPS 2018*.**

Dataset

For training all components, we used Korean text-speech dataset provided by AIHub that provides ~1,000 hours of Korean speech data in wav file format, paired with ETRI-format text.

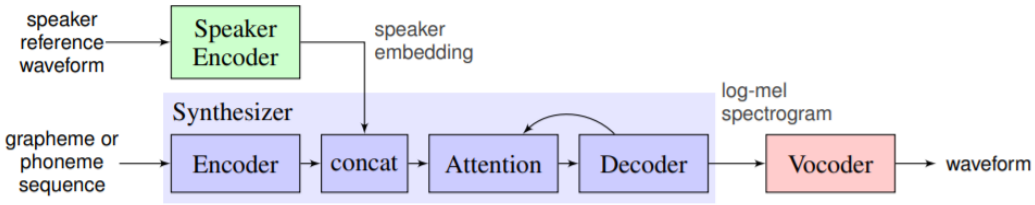
Preprocessing

We preprocessed text scripts in our dataset as follows. All parsing & preprocessing code were implemented by ourselves.

- We filtered special characters only used in ETRI transcription rule. After that, we converted every non-Korean characters (numbers, English) to Korean ("Model" > "모델").
- After preprocessing, we gathered all character occurrences in the dataset (가, 나, 다...) to make a character set representing input text in our TTS synthesis network.

Model Description

We followed the architecture and methodology of the reference paper [1]. Our model consists of three modules; speaker encoder network, seq2seq synthesis network and autoregressive vocoder network. Each network was trained separately.



- **Speaker encoder network**

  To condition TTS synthesis on speaker reference, the speaker encoder network generates a 256-dim speaker embedding vector from several seconds of voice sample. In [1], this was achieved by pretraining a stacked LSTM network on speaker identification task. However, our dataset had a critical problem; it lacked speaker identity label due to privacy issue, thus disallowing such transfer learning scheme. We tried to solve this problem with two different approaches.

  1. **Unsupervised approach** (implemented by ourselves)
     Our first idea was to train an autoencoder unsupervisedly from scratch so that we can use compressed representation as a speaker embedding [3].
     Following [3], we assumed that a joint representation connecting temporally-near speech segments can encode speaker identity. Thus, we trained the autoencoder to reconstruct a speech segment from another, temporally-near segment.
  2. **Pretrained for different language** (existing repo)
     Second idea was to use speaker encoder network pretrained on English speaker identification task, assuming that phonetic features of distinct speakers are independent of spoken languages. We cloned English-pretrained speaker encoder network from implementation of [1], and checked whether it generates Korean speaker embeddings that distinguishes different speaker groups.

- **Seq2seq synthesis network** (existing repo, modified and trained from scratch)

  Our TTS synthesizer network was based on Tacotron2 architecture, composed of an encoder and a decoder with attention. An input text is embedded through LSTM encoder, and encoder output is used by LSTM decoder to make a mel spectrogram.
  We modified the Tacotron2 model in two ways. First, we changed its input language from English to Korean to train on our preprocessed Korean text-speech dataset. More importantly, to condition the TTS synthesis on given speaker identity, we modified Tacotron2 code so that speaker embedding (generated by speaker encoder network) was concatenated with encoder output before going into the decoder. After that, we trained the network on our dataset from scratch.

- **Autoregressive vocoder network** (existing repo, fine-tuned)

  Our vocoder module was based on WaveRNN. We didn't modify its architecture, but since WaveRNN is pretrained with English speech data, it is not optimized to make Korean speaking sounds. Thus, we used our Korean dataset to fine-tune WaveRNN.

Training

~600,000 data in total, validation ratio 1%.

Optimizer: Adam		**Learning rate**	**Batch size**	**# iterations**
Speaker encoder	**Unsupervised (unused)**	1e-3	512	60k (~4 days)

	Pretrained (Used)	-	-	-
Synthesizer		1e-3	8	40k (~5 days)
Vocoder (fine-tuning)		1e-4	64	5k (~2 days)

Results

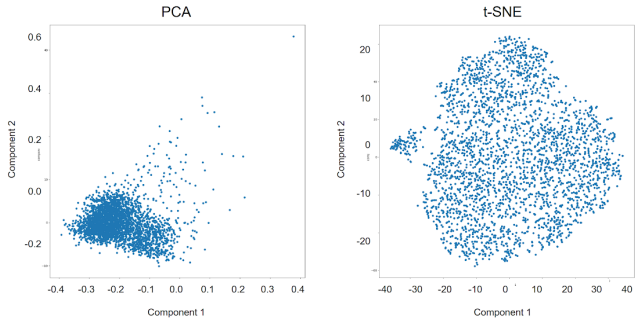


Figure 1. PCA (left) and t-SNE (right) plot of speaker embeddings generated by **autoencoder-based speaker encoder**. Embeddings are poorly distinguishable; speakers of distinct groups (e.g. male vs. females) are not clustered clearly.

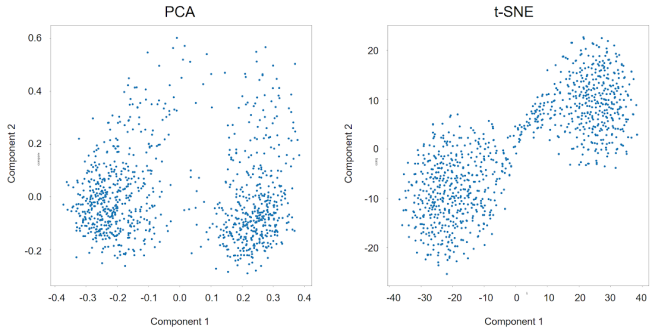


Figure 2. PCA (left) and t-SNE (right) plot of speaker embedding vectors generated by **speaker encoder pretrained on English speaker identification**. At least two significant clusters can be observed, presumably of male and female classes.

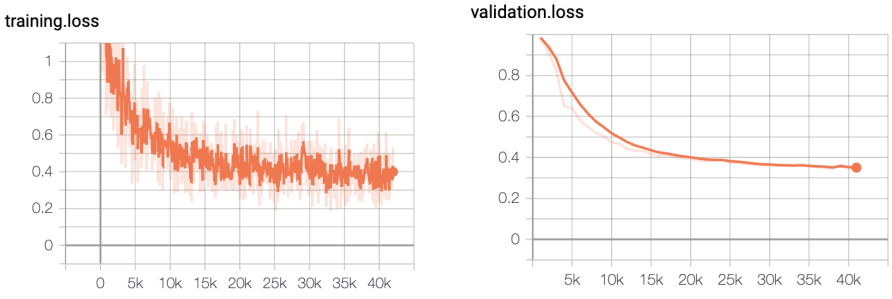


Figure 3. Training & validation curve of **speaker-conditioned synthesis network**.

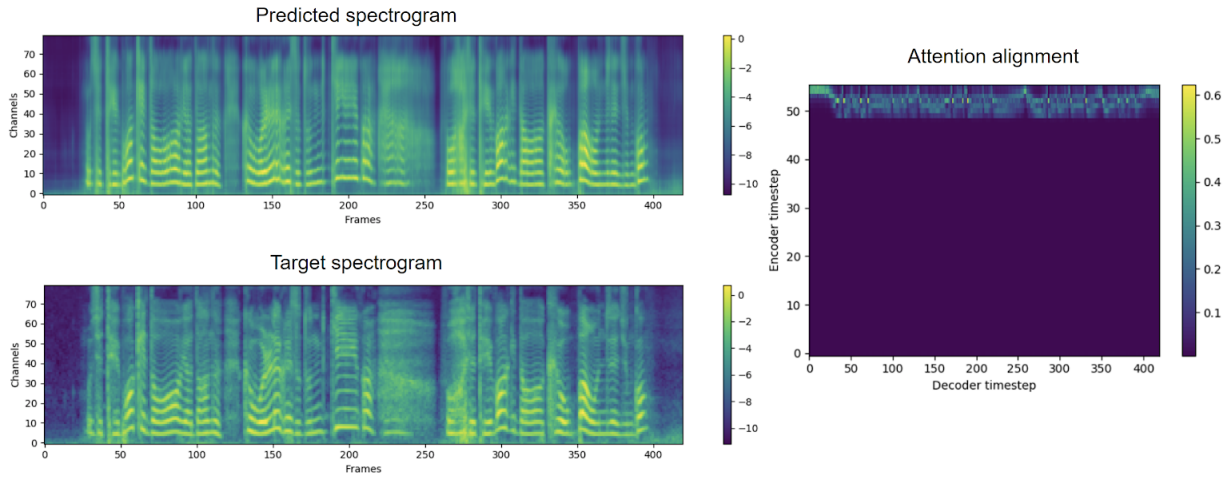


Figure 4. Attention alignment plot and target/generated mel spectrogram for a text input **during validation** of 38k-iteration synthesis network.

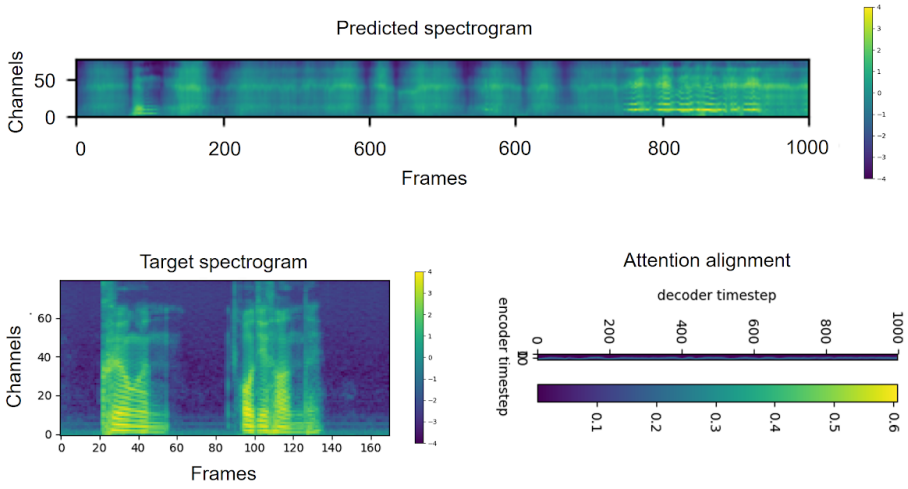


Figure 5. Attention alignment plot and target/generated mel spectrogram of 38k-iteration synthesis network for a **novel text input** “가나다라마바사아자차카파타하”.

Conclusion

- To solve unavailability of speaker identity labels in dataset, we first constructed an **unsupervised autoencoder-based model** to extract speaker identity from audio data. However, it turned out that the unsupervised model was unable to encode speaker identity effectively (Figure 1).
- After that, interestingly, we found that **speaker encoder network pretrained on English speaker identification task** generates distinct clusters of Korean speech embeddings (Figure 2), indicating that phonetic features that distinguishes different speakers do not depend on language.
- We **integrated our speaker embedding network into seq2seq (tacotron2) TTS model.** However, after 40k learning iterations (Figure 3, 4), the model was unable to

generate spectrogram conditioned to reference voice nor clear attention alignment for novel data (Figure 5). We list the possible reasons of failure below:

- **Small batch size** due to GPU memory limit
- **Short training** (case: 240k iter needed for convergence of vanilla tacotron2)
- Not using **silence trimming** to target spectrogram (hinders learning attention)

- We also suspect that **tokenizing Korean text further into syllables** (onset / nucleus / coda: 간 > ㄱ, ㅏ, ㄴ) is critical, as they directly encode phonetic information. As this encoding uses much more compact token set, we can also increase batch size without using more GPU memory. We implemented further tokenization into syllables and are using this to train a new model, but due to limited training time, we did not observe convergence yet.

Contribution

Sangwoo Kim (20150144) preprocessed the text transcripts and audio files / modified tacotron2 to run on the Korean sample data / trained synthesizer / evaluated synthesizer output / incorporated speaker encoder network into tacotron2 synthesizer / processed mel-spectrogram output of tacotron2 to run on vocoder / wrote the outline, speaker encoder and synthesizer module part of the final report / resolved dependency problem

Jaeyoon Kim (20193138) set up the training server / processed .wav data into mel and quant for vocoder finetuning / implemented and trained vocoder module / wrote vocoder module part in the final report.

Jinwoo Kim (20160171) implemented, trained and evaluated unsupervised speaker encoder / evaluated pretrained speaker encoder / incorporated speaker encoder into tacotron2 synthesizer / visualized mel spectrograms generated by synthesizer / adjusted audio sampling rate of vocoder output / implemented further decomposition of training text data into syllables and applied it to synthesizer training / did overall revision of the report / wrote GitHub documentation / wrote proposal, results and conclusion of the report.

Jinchul Ha (20193649) preprocessed text transcripts and audio files / modified tacotron2 to run on Korean sample data / trained synthesizer / evaluated synthesizer output / processed mel-spectrogram output of tacotron2 to run on the vocoder / pipelined all modules to work / wrote dataset and synthesizer module parts in the final report / resolved dependency problem

References

[1] Ye Jia et al., ‘Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis’. NIPS 2018.

[2] Kalchbrenner, Nal, et al., ‘Efficient Neural Audio Synthesis’. International Conference on Machine Learning. 2018.

[3] Arindam Jati et al., ‘Speaker2Vec: Unsupervised Learning and Adaptation of a Speaker Manifold using Deep Neural Networks with an Evaluation on Speaker Segmentation’. INTERSPEECH 2017.