Unsupervised Tex-to-Speech Synthesis by Unsupervised Automatic Speech Recognition

Junrui Ni, Liming Wang, Heting Gao

May 6, 2022
Outline

Motivation

Method

Unsupervised TTS on English

Unsupervised TTS on six other languages
Motivation

- Text-to-speech (TTS) synthesis is an essential component of a spoken dialogue system.
- Existing state-of-the-art TTS systems such as Tacotron 1&2, FastSpeech and Transformer TTS are trained with paired speech and text;
- Training a supervised text-to-speech (TTS) system requires dozens of hours of single-speaker high-quality recordings, which can be quite time-consuming and expensive to collect.

---


3N.Li et al. “Neural speech synthesis with transformer network”. In: AAAI. vol. 33. 2019, pp. 6706–6713
Outline

Motivation

Method

Unsupervised TTS on English

Unsupervised TTS on six other languages
Unsupervised TTS: problem formulation

- **Text input**: phoneme or character sequence $Y = [y_1, \cdots, y_m]$;
- **Speech input**: *Unpaired* speech $X = [x_1, \cdots, x_n], m \neq n$;
- **Output**: A generator function $G(\cdot)$ to map text into its corresponding speech waveform
Proposed model for unsupervised TTS

▶ Step 1: Learn an unsupervised ASR (wav2vec-U⁴) to generate pseudo-transcripts for $x_i$’s as $Y = [\tilde{y}_1, \cdots, \tilde{y}_n]$

---

Proposed model for unsupervised TTS

Text preprocessing: optionally apply **grapheme-to-phoneme (G2P)** converter on the character sequence
Proposed model for unsupervised TTS

- **wav2vec 2.0 and segment processing**: wav2vec 2.0 trained with LibriLight + PCA + average over consecutive segments assigned to the same **K-means clusters**
Proposed model for unsupervised TTS

- **CNN generator**: 1-layer CNN that outputs a sequence of distributions over text units where consecutive segments with the same arg max value are **collapsed**
Proposed model for unsupervised TTS

- **CNN discriminator**: 4-layer CNN that tries to tell which source (real or generated) the input sequence is from against the generator
Proposed model for unsupervised TTS

▶ Step 2: Learn a supervised TTS (Tacotron 2\(^5\)) to generate speech from pseudo-transcripts, \(\tilde{x}_i = G(\tilde{y}_i), i = 1, \cdots, n\)

\(^5\)Jonathan Shen et al. “Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions”. In: ICASSP. 2018
Proposed model for unsupervised TTS

▶ Tacotron 2 TTS: outputs **mel spectrograms**; follows the original Tacotron 2 model with additional guided attention loss \(^6\) to ensure that the attention matrix close to diagonal

\(^6\)Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. “Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention”. In: *ICASSP*. 2018, pp. 4784–4788
Proposed model for unsupervised TTS

Inference: use ground truth transcripts as inputs
Proposed model for unsupervised TTS

Vocoders: convert mel spectrogram into speech waveform; HiFiGAN\(^7\) or Griffin-Lim vocoders both implemented in ESPnet

\(^7\) Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. “HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis”. In: Neural Information Processing Systems. 2020
Outline

Motivation

Method

Unsupervised TTS on English

Unsupervised TTS on six other languages
Experiment 1: Unsupervised TTS on English

- **Speech dataset:** 24-hour single-speaker LJSpeech corpus
- **Text dataset:** transcripts from the LibriSpeech corpus (unpaired with the speech, distribution mismatch)
- **Train-test split:** 300 utterances for validation and 500 utterances for testing; convert to phonemes using G2P\(^8\)
- **wav2vec-U training:**
  1. **Grid search** the best weights for the auxiliary losses of the wav2vec-U system, i.e., code penalty, gradient penalty, and smoothness weight; 150k steps with a batch size of 160;
  2. **Self-training (ST) with a triphone HMM** using Framewise wav2vec 2.0+PCA features as input and pseudo phone sequences transcribed by the wav2vec-U generator as targets
  3. **Further ST with wav2vec 2.0 model** using the pseudo character targets obtained from the above step, and the Connectionist Temporal Classification (CTC) loss
- **TTS training:** trained for 80 epochs; HiFiGAN vocoder
- **Evaluation:** character error rate (CER, lower is better) and word error rate (WER, lower is better) by feeding the synthesized speech to a supervised ASR

\(^8\)Kyubyong Park and Jongseok Kim. *g2pE*. https://github.com/Kyubyong/g2pE. 2019
Results

Table: Unsupervised TTS results on the LJSpeech dataset

<table>
<thead>
<tr>
<th>Language</th>
<th>Unsup ASR (PER)</th>
<th>Unsup TTS</th>
<th>Supervised TTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No ST</td>
<td>ST</td>
<td>CER</td>
</tr>
<tr>
<td>English</td>
<td>12.37</td>
<td>3.59</td>
<td>4.56</td>
</tr>
</tbody>
</table>

- wav2vec-U sensitive to hyperparameters
- ST reduces the phone error rate on the test set by 70% relative
- UnsupTTS performed comparably with the supervised TTS

Figure: Mel-spectrograms for ground truth (upper) and synthetic speech by the unsupervised TTS model (lower) for the English sentence “in being comparatively modern.”
Outline

Motivation

Method

Unsupervised TTS on English

Unsupervised TTS on six other languages
Experiment 2: Unsupervised TTS on six languages from CSS10

- **Speech dataset**: Japanese, Hungarian, Spanish, Finnish, German and Dutch from the CSS10 dataset, each with 15hr speech
- **Text dataset**: from the same CSS10 dataset with their paired relationship broken up
- **Data split**: 99 to 1, which gave about 50 to 100 utterances validation
- **wav2vec-U training**: the same English wav2vec 2.0 Large model to extract speech representations and the same training pipeline but with only one ST stage; experiment with both characters and phonemes
- **TTS training**: 80 epochs similar to the English system; results obtained using Griffin-Lim vocoder by default
Overall Results

Table: Unsupervised TTS results on the CSS10 dataset using English wav2vec 2.0 pretrained features

<table>
<thead>
<tr>
<th>Language</th>
<th>Unsup TTS CER</th>
<th>Unsup TTS WER</th>
<th>Supervised TTS CER</th>
<th>Supervised TTS WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Japanese</td>
<td>17.98</td>
<td>47.81</td>
<td>17.87</td>
<td>36.23</td>
</tr>
<tr>
<td>Hungarian</td>
<td>27.78</td>
<td>76.82</td>
<td>18.05</td>
<td>63.14</td>
</tr>
<tr>
<td>Spanish</td>
<td>23.03</td>
<td>55.52</td>
<td>18.19</td>
<td>36.74</td>
</tr>
<tr>
<td>Finnish</td>
<td>36.05</td>
<td>84.46</td>
<td>22.84</td>
<td>58.67</td>
</tr>
<tr>
<td>German</td>
<td>17.25</td>
<td>56.78</td>
<td>11.28</td>
<td>40.94</td>
</tr>
<tr>
<td>Dutch</td>
<td>53.01</td>
<td>89.41</td>
<td>34.53</td>
<td>76.71</td>
</tr>
</tbody>
</table>

- Self-training step still greatly reduces the error rates by 25% to 40% relative to all the languages
- The CERs of UnsupTTS within 5% absolute to the supervised TTS in all languages; Much larger gap for WER
- In the case of German, the TTS trained with pseudo transcripts achieves a lower CER compared to the unsupervised ASR system, suggesting the existence of internal mechanism by TTS to correct the noise in the pseudo-transcripts
Characters vs phonemes

Table: Effect of different text units on unsupervised TTS using Griffin-Lim vocoder

<table>
<thead>
<tr>
<th>Language</th>
<th>Phoneme CER</th>
<th>Phoneme WER</th>
<th>Grapheme CER</th>
<th>Grapheme WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hungarian</td>
<td>22.73</td>
<td>68.80</td>
<td>27.78</td>
<td>76.82</td>
</tr>
<tr>
<td>Finnish</td>
<td>27.58</td>
<td>67.87</td>
<td>36.05</td>
<td>84.46</td>
</tr>
<tr>
<td>Dutch</td>
<td>22.04</td>
<td>56.85</td>
<td>53.01</td>
<td>89.41</td>
</tr>
</tbody>
</table>

- Use LanguageNet G2P*
- Both phoneme and character-based wav2vec-U can be unstable to train
- Phoneme-based system, when converged, achieves lower CER and WER than character-based system

*Mark Hasegawa-Johnson et al. “Grapheme-to-Phoneme Transduction for Cross-Language ASR”. In: SLSP. 2020, pp. 3–19
## Griffin-Lim vs HiFiGAN vocoders

**Table:** The effect of different pretrained vocoders (Griffin-Lim, HiFiGAN) on unsupervised TTS results for LJSpeech and various languages from CSS10

<table>
<thead>
<tr>
<th>Language</th>
<th>Griffin-Lim</th>
<th>HiFiGAN</th>
<th>Griffin-Lim</th>
<th>HiFiGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CER</td>
<td>WER</td>
<td>CER</td>
<td>WER</td>
</tr>
<tr>
<td>English</td>
<td>5.02</td>
<td>12.83</td>
<td><strong>4.56</strong></td>
<td><strong>11.95</strong></td>
</tr>
<tr>
<td>Japanese</td>
<td><strong>17.98</strong></td>
<td><strong>47.81</strong></td>
<td>20.58</td>
<td>54.09</td>
</tr>
<tr>
<td>Hungarian</td>
<td>27.78</td>
<td>76.82</td>
<td><strong>26.92</strong></td>
<td><strong>76.60</strong></td>
</tr>
<tr>
<td>Spanish</td>
<td><strong>23.03</strong></td>
<td><strong>55.52</strong></td>
<td>29.41</td>
<td>68.82</td>
</tr>
<tr>
<td>Finnish</td>
<td><strong>36.05</strong></td>
<td><strong>84.46</strong></td>
<td>37.66</td>
<td>87.48</td>
</tr>
<tr>
<td>German</td>
<td><strong>17.25</strong></td>
<td><strong>56.78</strong></td>
<td>18.45</td>
<td>59.90</td>
</tr>
</tbody>
</table>

- Griffin-Lim vocoder yields lower error rates than HiFiGAN on 4 out of 5 CSS10 languages.
- HiFiGAN yields lower error rate in English and produces more natural speech, but tend to skip phonemes on unseen languages.
Demo

TTS demo
Conclusion

- Propose UnsupTTS, an effective approach for unsupervised TTS
- Future direction: make wav2vec-U more stable; consider unmatched setting for multilingual data