VQVAE Unsupervised Unit Discovery and Multi-scale Code2Spec Inverter for Zerospeech Challenge 2019

Andros Tjandra¹,², Berrak Sisman³, Mingyang Zhang³, Sakriani Sakti¹,², Haizhou Li³, Satoshi Nakamura¹,²

¹ Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Japan
² RIKEN, Center for Advanced Intelligence Project AIP (RIKEN AIP), Japan
³ Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore
Outline

• Background
• Previous Works
• Proposed Approach:
  → Vector Quantized Variational Autoencoder (VQVAE)
  → Codebook-to-Spectrogram Inverter (Code2Spec)
• Experiments
• Conclusion
Background

• The ZeroSpeech 2019 challenge:
  Confronts the problem of constructing a speech synthesizer without any text or phonetic labels: TTS without T

• Two objectives:
  → Discover subword units in an unsupervised way
     (Encodes them as efficient as possible -- low-bitrate)
  → Using the encoded representation, synthesize the speech to a different target speaker
Previous Works

• Top performance in ZeroSpeech 2015 & 2017:
  → Unsupervised clustering with DPGMM [Chen et al., 2015; Heck et al. 2017]
  → But, DPGMM is too sensitive to acoustic variations [Wu et al., 2018]
  → It is difficult to synthesize speech from DPGMM-based unit

Achieving the best trade-off between unit discovery and speech synthesize is necessary
Proposed Method

• **Machine Speech Chain** [Tjandra et al., 2017]:
  - Enables ASR and TTS to assist each other when they receive unpaired data
  - Optimize both models with reconstruction loss

• **Inspired by a similar idea, we propose:**
  - Frame-based vector quantized variational autoencoder (VQ-VAE)
  - Multi-scale codebook-to-spectrogram inverter (Code2Spec)
Vector Quantized Variational Autoencoder (VQVAE)

- **Three Components of VQ-VAE** [van den Oord et al. 2017]
  - **Encoder** $q_\theta(z|x)$
    Read speech features $x \in \mathbb{R}^D$ and output latent variable $z \in \{1..K\}$
  - **Codebook** $E = [e_1, \ldots, e_K] \in \mathbb{R}^{K \times D_e}$
    Discretization is done by choosing the closest codebook
    $$q_\theta(z = c|x) = \begin{cases} 1 & \text{if } c = \arg\min_i \|\hat{z} - e_i\|_2 \\ 0 & \text{else} \end{cases}$$
    $$e_c = \sum_{i=1}^K q_\theta(z = i|x) e_i$$
  - **Decoder** $p_\phi(x|z, s)$ reconstruct the speech features conditioned by codebook $z$ and speaker $s$
    $$p_\phi(x|z, s) = p_\phi(x|e_c, v_s)$$
Vector Quantized Variational Autoencoder (VQVAE)

• **Training Objective:**

\[ \mathcal{L}_{VQ} = -\log p_{\phi}(x|z, s) + \|\text{sg}(\hat{z}) - e_c\|_2^2 + \gamma \|\hat{z} - \text{sg}(e_c)\|_2^2 \]

1. **Reconstruction loss** between speech and generated speech → To optimize encoder and decoder parameter
2. **Embedding Loss** or the update loss of the codebook dictionary → To optimize the move of embedding to the encoder output
3. **Consistency Loss** → As the volume of the embedding space is dimensionless, this loss is to forces encoder to generate a representation near the codebook

[Source: https://www.slideshare.net/fukuabca/vqvae]
Codebook-to-Spectrogram (Code2Spec)

- **Code2Spec Inverter Module**
  - Generates magnitude linear spectrogram given the codebook using multiscale 1D convolution
    \[
    \hat{M} = \text{Code2Spec}([e[1], e[1], \ldots, e[T_z], e[T_z]])
    \]
  - Training Objective:
    1. *MSE as reconstruction loss*
      \[\mathcal{L}_{MSE} = \|M - \hat{M}\|^2_2\]
    2. *GAN as auxiliary loss*
      \[
      \mathcal{L}_{GAN}^G = \begin{cases} 
      -\text{Disc}(\hat{M}) & \text{WGAN [Arjovsky et al., 2017]} \\
      (\text{Disc}(\hat{M}) - 1)^2 & \text{LSGAN [Mao et al., 2017]}
      \end{cases}
      \]
      \[
      \mathcal{L}_{GAN}^D = \begin{cases} 
      \text{Disc}(\hat{M}) - \text{Disc}(M) & \text{WGAN} \\
      \text{Disc}(M)^2 + (\text{Disc}(M) - 1)^2 & \text{LSGAN}
      \end{cases}
      \]
  - Waveform Generation:
    Reconstruct the missing phase with Griffin-Lim algorithm & apply STFT
Experimental Set-up

• **Dataset:** Default ZeroSpeech 2019 Data on English & Surprise Languages

• **Feature Extraction:**
  → Mel-spectrogram (80 dimensions, 25-ms window size, 10-ms time-steps)
  → MFCC (13 dims + Δ + Δ^2)

• **Feature representations:**
  → Directly using features (no model involves)
  → K-Means
  → GMM with diagonal covariances
  → VQ-VAE

• **Stride size to reduce the time length:**
  → Stride size: 1, 2, 4, 8
Results: Baseline & Experiment on Direct Features

• Baseline & Topline from ZeroSpeech

<table>
<thead>
<tr>
<th>Feature</th>
<th>ABX</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>35.63</td>
<td>71.98</td>
</tr>
<tr>
<td>Topline</td>
<td>29.85</td>
<td>37.73</td>
</tr>
</tbody>
</table>

• Direct feature representation (ABX with DTW cosine distance)

<table>
<thead>
<tr>
<th>Feature</th>
<th>ABX</th>
<th>Bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mel-Spec</td>
<td>30.291</td>
<td>1738.38</td>
</tr>
<tr>
<td>MFCC</td>
<td>21.114</td>
<td>1737.47</td>
</tr>
</tbody>
</table>

MFCC produced better performances on the ABX metric than the Melspectrogram. But, the bit rate still remains too high.
Results: K-Means, GMM & Proposed VQ-VAE

• **K-Means continuous representation**

<table>
<thead>
<tr>
<th>Model</th>
<th>#C</th>
<th>1T</th>
<th>2T</th>
<th>4T</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-Means (cont, DTW cos)</td>
<td>64</td>
<td>23.56 / 553</td>
<td>25.97 / 280</td>
<td>29.41 / 136</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>23.16 / 649</td>
<td>24.24 / 321</td>
<td>28.12 / 161</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>21.90 / 744</td>
<td>23.73 / 369</td>
<td>27.17 / 182</td>
</tr>
</tbody>
</table>

• **GMM posterior representation**

<table>
<thead>
<tr>
<th>Model</th>
<th>#C</th>
<th>1T</th>
<th>2T</th>
<th>4T</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM (post, DTW KL)</td>
<td>64</td>
<td>20.81 / 1647</td>
<td>22.67 / 676</td>
<td>29.82 / 257</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>19.61 / 1705</td>
<td>23.06 / 704</td>
<td>31.19 / 281</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>18.93 / 1691</td>
<td>23.39 / 757</td>
<td>32.99 / 306</td>
</tr>
</tbody>
</table>

• **Proposed Approach: VQ-VAE codebook representation**

<table>
<thead>
<tr>
<th>Model</th>
<th>#CL</th>
<th>1T</th>
<th>2T</th>
<th>4T</th>
<th>8T</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQ-VAE (cont, DTW cos)</td>
<td>64</td>
<td>27.46 / 606</td>
<td>25.51 / 302</td>
<td>26.15 / 138</td>
<td>28.81 / 70</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>27.65 / 686</td>
<td>24.29 / 347</td>
<td>25.04 / 165</td>
<td>30.87 / 79</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>27.63 / 787</td>
<td>24.37 / 349</td>
<td>24.17 / 184</td>
<td>30.51 / 79</td>
</tr>
<tr>
<td></td>
<td>512</td>
<td>27.69 / 871</td>
<td>23.59 / 400</td>
<td>24.63 / 180</td>
<td>32.02 / 74</td>
</tr>
</tbody>
</table>

VQ-VAE model has the best tradeoff between ABX and bit-rate compared to K-Means, GMM and direct MFCC features.
Conclusion

• **VQ-VAE model has the best tradeoff between ABX and bit-rate compared to K-Means, GMM and direct MFCC features**

• **Things we tried but didn’t work well:**
  → Wavenet vocoder
    The codebook every 20 or 40 ms perhaps too sparse
  → **GAN speech enhancement**
    Effective for achieving high-quality VC with clean speech, not for distorted speech

• **Our best submission:**
  → VQ-VAE+Code2Spec with 256 codebooks and 2 & 4 time-stride
  → Significantly improved performance from baseline (even the topline):
    the intelligibility - CER (**Rank 1st**), the naturalness - MOS (**Rank 3rd**),
    and the discrimination - ABX scores (**Rank 4th**)

Copyright @ AHC Lab NAIST - 2019
Sequel: S2ST without T

Andros Tjandra, Sakriani Sakti, Satoshi Nakamura, "SPEECH-TO-SPEECH TRANSLATION BETWEEN UNTRANSCRIBED UNKNOWN LANGUAGES," ASRU, pp. to appear, 2019
Thank you