On the Usage of Phonetic Information for Text-independent Speaker Embedding Extraction

Shuai Wang\textsuperscript{1,2}, Johan Rohdin\textsuperscript{2}, Lukáš Burget\textsuperscript{2}, Oldřich Plchot\textsuperscript{2}, Yanmin Qian\textsuperscript{1}, Kai Yu\textsuperscript{1}, Jan Černocký\textsuperscript{2}

\textsuperscript{1}Speech Lab, Shanghai Jiao Tong University, China
\textsuperscript{2}Speech@FIT, Brno University of Technology, Czechia

September 2019
Background

- The development of ASR techniques has greatly inspired the SID community.
- Researchers investigated to incorporate phonetic information for speaker embedding learning.

Basic assumption for this paper:
Good text-independent speaker embeddings are not expected to be affected by spoken content, it might be helpful to explicitly suppress the phonetic variability in the final embeddings.
Explicitly modeling phonetic information helps the text-dependent speaker verification task, which is intuitive.

Related work
multi-task learning in the x-vector framework

Why explicitly learning phonetic information helps the text-independent speaker verification task? Is it counter-intuitive?

- **Text-independent task**
- Multi-task at the frame-level
- Performance improved!

---

Related work
Speaker invariant training for ASR

- Acoustic modelling
- Adversarial training suppressing the speaker effect
- Performance improved

---

What will happen if we train a speaker classifier, while making it unable to discriminate different phoneme classes?
Adversarial training
Explicitly suppress the phonetic information in speaker embedding

- Consider speaker classification as the primary task and phoneme classification as the secondary task.
- Multi-task aims to minimize the classification loss of both tasks.
- In adversarial training, the goal is to minimize the speaker classification loss and **mini-maximize** the phoneme classification loss.
Frame-level multi-task/adversarial training

Speaker Loss

- Speaker classifier $M_s$
- Segment level
- Dense Layer
- Dense Layer
- Statistics Pooling

Phoneme Loss

- Phoneme classifier $M_p$
- Frame level
- Dense Layer
- Dense Layer
- Gradient Reversal

TDNN Feature Extractor $M_f$

Frame-level

input segment X

\[ \mathcal{L}_s = \text{CE}(M_s(M_f(X)), y^s) \]

\[ \mathcal{L}_p = \frac{1}{N} \sum_{i=1}^{N} \text{CE}(M_p(M_f(x_i)), y^p_i) \]

\[ \mathcal{L}_{total} = \mathcal{L}_s + \mathcal{L}_p \]
However, we observe a big performance degradation for the frame-level adversarial training

- Granularity might be the problem, it’s harder to remove fine-grained information from coarse-grained information
- What will happen if we do both tasks at the same granularity?
Segment-level multi-task/adversarial training

\[ \mathcal{L}_s = \text{CE}(M_s(M_f(X)), y^s) \]
\[ \mathcal{L}_p = \text{CE}(M_p(M_f(x_i)), y^p) \]
\[ \mathcal{L}_{total} = \mathcal{L}_s + \mathcal{L}_p \]

Question: How to represent \( y^p \)?
For a given segment $x$ with $N$ frames, the corresponding segment-level phoneme label $y^p$ is represented as

$$y^p = \{y_1, y_2, \ldots, y_C\}$$

$$y_c = \frac{N_c}{N}$$

where $C$ is the size of the chosen phoneme set. $N_c$ denotes the number of occurrences of the $c$-th phoneme in $x$. 
Experiments
Setups

**Dataset**

**Training data:**
Voxceleb1 Dev + Voxceleb2 Dev

**Evaluation data:**
Voxceleb1 Eval

**Speaker Embedding Extractor**
All speaker embedding systems are based on the TDNN x-vector

**Phoneme recognizer**
- The phoneme labels are generated from a phoneme recognizer
- 166 classes: position-dependent phonemes + SIL and NOISE nodes
- Training follows official Kaldi Tedlium speech recognition recipe
### Experiments

Results: Frame-level multi-task/adversarial training

Table: Systems combining frame-level phonetic information, FRM-MT and FRM-ADV denote two systems trained using multitask or adversarial objectives, with or without the gradient reversal layer, respectively.

<table>
<thead>
<tr>
<th>System</th>
<th>EER(%)</th>
<th>minDCF$_{0.1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-vector baseline</td>
<td>3.73</td>
<td>0.192</td>
</tr>
<tr>
<td>FRM-MT</td>
<td>3.38</td>
<td>0.180</td>
</tr>
<tr>
<td>FRM-ADV</td>
<td>5.24</td>
<td>0.269</td>
</tr>
</tbody>
</table>
Experiments

Results: Segment-level multi-task/adversarial training

Table: Systems combining segment-level phonetic information, SEG-MT and SEG-ADV denote two systems trained using multitask or adversarial objectives, with or without the gradient reversal layer, respectively

<table>
<thead>
<tr>
<th>System</th>
<th>EER(%)</th>
<th>minDCF$_{0.1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-vector baseline</td>
<td>3.73</td>
<td>0.192</td>
</tr>
<tr>
<td>SEG-MT</td>
<td>3.71</td>
<td>0.175</td>
</tr>
<tr>
<td>SEG-ADV</td>
<td>3.35</td>
<td>0.159</td>
</tr>
</tbody>
</table>
Experiments

Results: Combining frame-level multitask and segment-level adversarial learning

Table: Systems combining frame-level multitask and segment-level adversarial learning

<table>
<thead>
<tr>
<th>System</th>
<th>EER(%)</th>
<th>minDCF&lt;sub&gt;0.1&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-vector baseline</td>
<td>3.73</td>
<td>0.192</td>
</tr>
<tr>
<td>FRM-MT</td>
<td>3.38</td>
<td>0.180</td>
</tr>
<tr>
<td>SEG-ADV</td>
<td>3.35</td>
<td>0.159</td>
</tr>
<tr>
<td>FRM-MT + SEG-ADV</td>
<td>3.17</td>
<td>0.163</td>
</tr>
</tbody>
</table>
Conclusions

- **Main Contribution**
  - The architecture of training multi-task/adversarial systems at the segment level
  - Experiments to examine the impact of phonetic information on text-independent speaker embedding learning

- Our experiments show that for text-independent speaker embedding learning, it’s beneficial to
  - enhance the fine-grained phonetic information at the frame-level part
  - suppress phonetic information at the segment-level part