Improved Language Identification Through Cross-Lingual Self-Supervised Learning

Andros Tjandra, Diptanu Gon Choudhury, Frank Zhang, Kritika Singh, Alexis Conneau, Alexei Baevski, Assaf Sela, Yatharth Saraf, Michael Auli
Outline

1. Background
2. Log-Mel Wav2Vec
3. Cross-lingual Speech Representation (XLSR)
4. LID Finetuning
5. Experimental setup
6. Results
7. Conclusion
Background

• Language identification (LID) predicts which language is being spoken given a speech utterance and routes the speech to the correct ASR service.

• Problem & prior works:
  - Modern LID are trained with large amounts of labeled data.
  - Self-supervised learning (e.g., Wav2Vec 2.0) can leverage unlabeled data.
  - Prior study [1] shows Wav2Vec 2.0 trained on English data could improve LID.

• Proposed:
  - Extend the study using cross-lingual self-supervised (XLSR) to improve LID accuracy.
  - Explore different 1) content aggregation strategies, 2) pruning layer size.

[1] Exploring wav2vec 2.0 on Speaker Verification and Language Identification, Fan et al. Interspeech 2021
Wav2Vec 2.0

- Wav2Vec 2.0 is a self-supervised learning trained with audio only.
- It consists of 3 main components
  - Feature encoder: extract latent speech representation $Z$
  - Quantizer: learn contextualized representation $Q$ from continuous $Z$
  - Context encoder: learn high-level speech representation
- Loss function:
  \[
  \mathcal{L}_m = -\log \frac{\exp\left(\frac{\text{sim}(c_t, q_t)}{\kappa}\right)}{\sum_{\tilde{q} \sim Q_t} \exp\left(\frac{\text{sim}(c_t, \tilde{q})}{\kappa}\right)}
  \]
Log-Mel Wav2Vec

- Instead of using raw-waveform, we replace the input feature into log-mel spectrogram.
- We also modify the Wav2Vec architecture by replacing stack of 1D convolution layers with time-stacking + a linear layer.
- By using this modification, we could reduce the memory usage and improve running time to support large-scale training more efficiently.
Cross-lingual Speech Representation (XLSR)

- XLSR is a multilingual Wav2Vec 2.0, trained altogether with various unlabeled speech from different languages.
- In this paper, since we don’t have any language metadata on the unlabeled speech, we train our XLSR without any data rebalancing.
LID Finetuning

- We initialized the bottom part of the LID classifier with a pre-trained Wav2Vec.
- We explored several pooling operations such as:
  - Mean pooling \( o = \frac{\sum_{t=1}^{T} c_t}{T} \)
  - Std. dev pooling \( o = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (c_t - \mu)^2} \)
  - Self-attention pooling
    \[
    o = \sum_{t=1}^{T} \text{softmax} \left( w_2 \text{GELU} \left( W_1 c^T \right) \right) c_t
    \]
  - Concat [CLS] token in the 1\(^{st}\) index and set \( o = c_1 \)
- Loss criterion: cross-entropy
Experimental Setup

• Pre-training XLSR
  - We are using 6.3 million hours unlabeled speech with different languages and conditions (e.g., clean, noisy, etc).
  - Log-mel Wav2Vec configuration:
    - Feature encoder: 4x time-stack stride + linear layer
    - Context encoder: 24 Transformers layer ($d_{in} = 1024, d_{ffn}=4096$)
    - Quantization module: Gumbel VQ (320 codebooks, 2 groups)

• Finetuning
  - We finetune with LID dataset consisted of 26 languages
Experimental Setup (cont.)

- Evaluation
  - Inference with 6 seconds audio & 3 seconds step size and we average the language probability across multiple segments to predict the class.

\[ p = \text{avg}(p_0, p_1, \ldots, p_{S/3}) \]
Exp 1: LID accuracy in different setups

- We compare 3 different models: 1) without pre-training, 2) pre-trained with Wav2Vec English data, 3) XLSR
- We also compare the accuracy between different amount of labeled data, test data lengths.
- XLSR shows best performance overall.

<table>
<thead>
<tr>
<th>Lbl. / lang</th>
<th>Pre-training</th>
<th>Test Accuracy (%)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0-6s</td>
<td>6-18s</td>
<td>18-∞s</td>
<td>Overall</td>
<td></td>
</tr>
<tr>
<td>10 min</td>
<td>None</td>
<td>7.1</td>
<td>9.5</td>
<td>10.6</td>
<td>9.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w2v2 En</td>
<td>71.3</td>
<td>73.1</td>
<td>76.1</td>
<td>74.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XLSR</td>
<td>85.4</td>
<td>88.8</td>
<td>90.8</td>
<td>89.2</td>
<td></td>
</tr>
<tr>
<td>1 hour</td>
<td>None</td>
<td>20.2</td>
<td>25.2</td>
<td>29.5</td>
<td>26.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w2v2 En</td>
<td>79.3</td>
<td>85.9</td>
<td>89.3</td>
<td>86.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XLSR</td>
<td>87.2</td>
<td>92.5</td>
<td>94.8</td>
<td>92.8</td>
<td></td>
</tr>
<tr>
<td>10 hours</td>
<td>None</td>
<td>48.3</td>
<td>61.9</td>
<td>71.8</td>
<td>64.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w2v2 En</td>
<td>86.8</td>
<td>93.3</td>
<td>95.6</td>
<td>93.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XLSR</td>
<td>88.2</td>
<td>94.3</td>
<td>96.1</td>
<td>94.2</td>
<td></td>
</tr>
<tr>
<td>100 hours</td>
<td>None</td>
<td>72.2</td>
<td>84.9</td>
<td>90.7</td>
<td>86.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w2v2 En</td>
<td>89.5</td>
<td>95.7</td>
<td>97.3</td>
<td>95.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XLSR</td>
<td>90.3</td>
<td>95.9</td>
<td>97.2</td>
<td>95.7</td>
<td></td>
</tr>
</tbody>
</table>
Exp 2: Pruning context Transformers layers

- We tried to prune the layers from context encoders.
- Our results show that we could prune up to 2/3 layers (reduce params from 300m -> 100m) and still maintain the same accuracy.
Exp 3: Pooling strategy

<table>
<thead>
<tr>
<th>Aggregation strategy</th>
<th>Accuracy (%)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0-6s</td>
<td>6-18s</td>
<td>18-(\infty)</td>
<td>Overall</td>
</tr>
<tr>
<td>Max</td>
<td>86.6</td>
<td>92.7</td>
<td>94.8</td>
<td>92.8</td>
</tr>
<tr>
<td>Mean+Max+Min</td>
<td>88.1</td>
<td>92.9</td>
<td>94.7</td>
<td>93.0</td>
</tr>
<tr>
<td>Mean+Max</td>
<td>88.5</td>
<td>93.1</td>
<td>94.8</td>
<td>93.2</td>
</tr>
<tr>
<td>Mean+Std</td>
<td>84.2</td>
<td>90.9</td>
<td>93.4</td>
<td>91.1</td>
</tr>
<tr>
<td>[CLS] Token</td>
<td>85.4</td>
<td>91.4</td>
<td>93.9</td>
<td>91.7</td>
</tr>
<tr>
<td>Self Attention</td>
<td>87.0</td>
<td>92.0</td>
<td>94.1</td>
<td>92.2</td>
</tr>
</tbody>
</table>

- We tried different pooling on 1hr/lang setup. Mean+Max pooling shows the best accuracy despite its simplicity.
Conclusion

• We showed that cross-lingual pre-trained model are particularly effective for low-resource setups, especially when little labeled data is available.
• Using only 10 minutes per languages, an XLSR-based LID could achieve 89.2% on 26 languages setup.
• From our pruning study, we could prune 2/3 of context encoder layers and still maintain its accuracy.
• Simple pooling strategy such as max+mean works well in this system.