A TRANSFER LEARNING APPROACH FOR PRONUNCIATION SCORING

Marcelo Sancinetti1, Jazmín Vidal1,2, Cynthia Bonomi2, Luciana Ferrer1
1 Laboratorio de Inteligencia Artificial Aplicada, Instituto de Ciencias de la Computación, UBA-CONICET, Argentina
2 Departamento de Computación, Universidad de Buenos Aires, Argentina

*jvidal@dc.uba.ar

PRONUNCIATION SCORING

Given a phrase uttered by a language learner, return a pronunciation quality score for each phone.

- Challenging task with room for improvement.
- Standard systems use models trained for automatic speech recognition (ASR) with native data only.
- Better performance using systems trained specifically for the task using native data.
- Datasets labelled for the task are scarce and usually small.

NATIVE DATA
- Rely on ASR technology to generate native models.
- Measures similarity between student’s speech and native sounding speech.

NATIVE = NONNATIVE DATA
- Use non-native data with pronunciation quality labels.
- Directly trained to distinguish correctly from incorrectly pronounced segments.
- Variety of input features and classifiers.

TRANSFER LEARNING
- DNNs for pronunciation scoring show improvements over traditional methods of both groups.
- Rely on transfer learning to mitigate data scarcity.

BASELINE METHOD: GOP

- GOP scores: for each phone, the averaged posterior probability of the target phone for each frame.
- Computed using the outputs of a senone acoustic model.

\[
GOP(p) = \frac{1}{D} \sum_{t=1}^{D} \log P_t(p|\theta)
\]

- Start and end frames are obtained from forced alignment.

EXPERIMENTS

- LayO: only the new output layer is trained, keeping all other parameters frozen at their pre-trained values.
- BN: batch-normalization in the output layer.
- DO: dropout in all layers.
- Bal: the loss with balanced weights is used in training.

RESULT / CONCLUSIONS

- Average 1-AUC and MinCost (phones with more than 50 samples of each class for the development data)
- GOP system has 1-AUC of 0.286 and MinCost of 0.801.
- Best configuration: LayO+1 BN DO BAL

COST FUNCTION

- Allows to control false negatives / useful for pedagogical reasons
- \[\text{Cost} = 0.5 \text{FPR} + \text{FNR}\]
- Allows to see the effect of the threshold selection
- MinCost: computed on test data / ActCost: computed on dev data.

CONTRIBUTIONS

- Fine-tune the ASR model to the task of pronunciation scoring.
- Explore 2 different fine-tuning approaches and 6 design choices.
- Propose a loss function that compensates for inherent imbalance across phones and classes present in pronunciation scoring datasets.
- Measure performance using an alternative cost function designed to encourage low false correction rates.
- Share dataset and code to replicate the results at:
  https://github.com/MarceloSancinetti/epa-gop-pykaldi

3200 nonnative English phrases by 50 speakers from Argentina.
- Manually annotated at detailed phonetic level using ARPA/tet symbols.
- Correctly- and incorrectly-pronounced labels are assigned to each of the target phones determined by the forced-alignment system.

DATABASE

- Official Kaldi recipe reproduced in Pykaldi
- Features: 40-dimensional MFCCs + 1-vectors.
- Acoustic model: TDNN-F trained on LibriSpeech (960 hours) (decoding and forced alignment)
- 17 layers + output layer of size 6024 senones + softmax

TOP BRANCH - GOP

Bottom branch - GOP-FT

- Official Kaldi recipe reproduced in Pykaldi
- Features: 40-dimensional MFCCs + 1-vectors.
- Acoustic model: TDNN-F trained on LibriSpeech (960 hours) ported to Pykaldi (decoding and forced alignment)
- 17 layers + affine output layer of size 39 + sigmoid

- Replace baseline output layer with a layer that predicts per phone per frame probability of correctly pronounced

- GOP-FT scores: for each frame the probability of being correctly pronounced for the target phone in that frame. Then average over the frames.

- Weighted cross-entropy loss:

\[
L = \sum_{t} \sum_{p} y_{tp} \log (y_{tp}) + (1-y_{tp}) \log (1-y_{tp})
\]

- Adjust the influence of the samples from each phone and class.
  - Flat Weights: \(w_p = 1\)
  - Zero Weight: \(w_p = 0\)
  - Balanced: \(w_p = 1/N_p\)

-GOP-FT: designed to encourage low false correction rates.
- Propose a loss function that compensates for inherent imbalance across phones and classes present in pronunciation scoring datasets.
- Measure performance using an alternative cost function designed to encourage low false correction rates.
- Share dataset and code to replicate the results at:
  https://github.com/MarceloSancinetti/epa-gop-pykaldi

- REPLACING THE OUTPUT LAYER
- GD: Gradients with backpropagation.
- **GO**P-FT: TDNN-F trained on LibriSpeech (960 hours) ported to Pykaldi (decoding and forced alignment)
- 17 layers + affine output layer of size 39 + sigmoid

- Replace baseline output layer with a layer that predicts per phone per frame probability of correctly pronounced

- GOP-FT scores: for each frame the probability of being correctly pronounced for the target phone in that frame. Then average over the frames.

- Weighted cross-entropy loss:

\[
L = \sum_{t} \sum_{p} y_{tp} \log (y_{tp}) + (1-y_{tp}) \log (1-y_{tp})
\]

- Adjust the influence of the samples from each phone and class.
  - Flat Weights: \(w_p = 1\)
  - Zero Weight: \(w_p = 0\)
  - Balanced: \(w_p = 1/N_p\)

- GOP-FT: designed to encourage low false correction rates.
- Propose a loss function that compensates for inherent imbalance across phones and classes present in pronunciation scoring datasets.
- Measure performance using an alternative cost function designed to encourage low false correction rates.
- Share dataset and code to replicate the results at:
  https://github.com/MarceloSancinetti/epa-gop-pykaldi

- REPLACING THE OUTPUT LAYER
- GD: Gradients with backpropagation.
- **GO**P-FT: TDNN-F trained on LibriSpeech (960 hours) ported to Pykaldi (decoding and forced alignment)
- 17 layers + affine output layer of size 39 + sigmoid

- Replace baseline output layer with a layer that predicts per phone per frame probability of correctly pronounced

- GOP-FT scores: for each frame the probability of being correctly pronounced for the target phone in that frame. Then average over the frames.

- Weighted cross-entropy loss:

\[
L = \sum_{t} \sum_{p} y_{tp} \log (y_{tp}) + (1-y_{tp}) \log (1-y_{tp})
\]

- Adjust the influence of the samples from each phone and class.
  - Flat Weights: \(w_p = 1\)
  - Zero Weight: \(w_p = 0\)
  - Balanced: \(w_p = 1/N_p\)

- GOP-FT: designed to encourage low false correction rates.
- Propose a loss function that compensates for inherent imbalance across phones and classes present in pronunciation scoring datasets.
- Measure performance using an alternative cost function designed to encourage low false correction rates.
- Share dataset and code to replicate the results at:
  https://github.com/MarceloSancinetti/epa-gop-pykaldi

- REPLACING THE OUTPUT LAYER
- GD: Gradients with backpropagation.
- **GO**P-FT: TDNN-F trained on LibriSpeech (960 hours) ported to Pykaldi (decoding and forced alignment)
- 17 layers + affine output layer of size 39 + sigmoid

- Replace baseline output layer with a layer that predicts per phone per frame probability of correctly pronounced

- GOP-FT scores: for each frame the probability of being correctly pronounced for the target phone in that frame. Then average over the frames.

- Weighted cross-entropy loss:

\[
L = \sum_{t} \sum_{p} y_{tp} \log (y_{tp}) + (1-y_{tp}) \log (1-y_{tp})
\]

- Adjust the influence of the samples from each phone and class.
  - Flat Weights: \(w_p = 1\)
  - Zero Weight: \(w_p = 0\)
  - Balanced: \(w_p = 1/N_p\)

- GOP-FT: designed to encourage low false correction rates.
- Propose a loss function that compensates for inherent imbalance across phones and classes present in pronunciation scoring datasets.
- Measure performance using an alternative cost function designed to encourage low false correction rates.
- Share dataset and code to replicate the results at:
  https://github.com/MarceloSancinetti/epa-gop-pykaldi