Monophone-based Background Modeling for two-stage on-device Wake Word Detection

Minhua Wu, Sankaran Panchapagesan, Ming Sun, Jiacheng Gu, Ryan Thomas, Shiv Naga Prasad Vitaladevuni, Bjorn Hoffmeister, Arindam Mandal

wuminhua@amazon.com, panchi@google.com, mingsun@amazon.com

Motivation

Goal: To improve the accuracy of the wake word detector on the Amazon device

Focus of this work: Incorporate monophone-based units to model the non-keyword background

Baseline Two-stage Wake Word System

1st Stage DNN-HMM Decoder

- Background HMM: speech/non-speech loop
- Foreground HMM: ACLA phone state sequence

- Features from candidates
- Final decision

2nd Stage Classifier

- Capture candidates from the whole candidate segment e.g. phone duration, confidence scores etc.
- Use a small feed-forward Neural Network (NN) for experiments

New Wake Word System Using Monophone-based Background Modeling

New 1st Stage DNN-HMM Decoder

- New Background HMM:
  - Expand speech, non-speech events to various monophones
  - Becomes a phone-level unigram FST
- New Acoustic Model: background targets expanded

New Feature Engineering for 2nd Stage Classifier

- Foreground HMM: wake word phone states
- Background HMM: speech and non-speech states loop
- Acoustic Model: Deep Neural Network (DNN)
- Decoder: Viterbi decoding on the graph
- Wake Word Hypothesized: When difference in foreground and background log likelihoods exceeds a threshold
- 1st stage DET curve: Tune weight on arcs and states

2nd Stage Classifier

- Second Stage Feature Vector
  - Obtained from 1st stage wake word hypothesis
  - Captures info from the whole candidate segment (e.g. segment duration, likelihood etc.)
  - Captures info related to phone segment (e.g. phone duration, confidence scores etc.)
- Use a small feed-forward Neural Network (NN) for experiments

Experiment Results

Baseline Setup

- Several thousand hours of real far-field data for training
- Approximately 30,000 wake word instances in dev/test set
- A feed-forward DNN acoustic model at the first stage
- Features: Log Mel-Filter-Bank Energies (FBFE) (20 frames for left context and 10 frames for right context)
- Wake word task (50 targets) multi-task trained with LVCSR targets [1]
- The GPU-based distributed DNN trainer utilized [2]
- The second stage feature vector is of dimension 67
- A small feedforward NN as the second stage classifier

Changes for Monophone-based System Setup

- Use 44 monophones in the background model
- Background HMM changes to be a phone-level unigram FST
- DNN output targets for the wake word task is expanded
- The second stage feature vector is of dimension 375 (67x94)

1st Stage HMM Tuning

- Performance: The two systems are almost the same at this stage
- Operating points picked for building 2nd stage classifier:
  - recall at around 0.02 for both systems

End-to-end Evaluation

Table 1: Summary of different wake word systems

<table>
<thead>
<tr>
<th>Model</th>
<th>FRR (Fix FAR=0.04)</th>
<th>FAR (Fix FRR=0.04)</th>
<th># of Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP/NSP(4x896)</td>
<td>0.051</td>
<td>3.71y</td>
<td>3.02M</td>
</tr>
<tr>
<td>SP/NSP(4x1024)</td>
<td>0.050</td>
<td>3.49y</td>
<td>3.94M</td>
</tr>
<tr>
<td>Monophone(4x896)</td>
<td>0.043</td>
<td>2.35y</td>
<td>3.15M</td>
</tr>
<tr>
<td>Monophone(4x1024)</td>
<td>0.042</td>
<td>2.31y</td>
<td>3.99M</td>
</tr>
</tbody>
</table>

Effectiveness of the 2nd stage

Figure 6: Comparison of the performance with and without 2nd stage classifier (2x94 NN). DET curves on test set: Axis of the false alarm rate is obscured due to the sensitive nature of this information

Conclusion

- Propose a new way to model the non-keyword part.
  - Expand the speech/non-speech events to more specific monophone-based units at the first stage.
  - Extract extra match scores for final detection
- The new system reduces FAR by 37% when the FRR level is maintained.
- On the other hand, it reduces FRR by about 16% if FAR level is fixed.
- The second stage itself is able to reduce FRR by 67% relatively on top of 1st stage hypothesis.

References

[2] Nikko Strom. Scalable distributed dnn training using commodity gpus