Unsupervised network adaptation and phonetically-oriented system combination for the CHiME-4 Challenge

09/13/2016

**Yusuke Fujita, Takeshi Homma, Masahito Togami**

Hitachi Ltd., Research and Development Group, Japan
Hitachi America Ltd.
• Hitachi is developing a humanoid robot "EMIEW3" for customer services (ex. airport, station, bank)
  – Distant (1m) ASR
  – Noise robustness in real fields is crucial
• We participated in CHiME-3 challenge
  – Local Gaussian modeling based source separation works well with DNN-based ASR
  – Discriminative system combination outperforms ROVER
  – However, data augmentation, speaker adaptation, and RNNLM examined by top teams have not been applied.
• We followed these state-of-the-art techniques and updated our system for CHiME-4
Contributions

• Local Gaussian modeling based source separation
  – Multi-channel Wiener filter output is utilized for acoustic modeling and frontend speech enhancement
  – Introducing semi-stationarity constraints to non-target sources improves frontend speech enhancement

• Unsupervised deep neural network adaptation
  – Unsupervised re-training of DNN works well for speaker adaptation when using conservative training parameters

• Phonetically-oriented system combination
  – Multiple 1-best sentences are combined considering phonetic similarity improves the system combination performance
Local Gaussian modeling based source separation

- Multi-channel signal in time-frequency domain
  \[ x(f, t) = [x_1(f, t), \cdots, x_M(f, t)]^\top \in \mathbb{C}^M \]
  f: frequency, t: time (frame), M: # microphones

- Local Gaussian modeling (LGM) [Duong et al. 2010]
  \[ x(f, t) = \sum_{n=1}^{N} c_n(f, t) \]
  - Spatial image of each source
  \[ c_n(f, t) \sim \mathcal{N}_{\mathbb{C}}(0, v_n(f, t)V_n(f)) \]
    time-variant activity  time-invariant spatial correlation matrix
Local Gaussian modeling based source separation

• Multi-channel Wiener filter (MCWF)
  \[ c_n(f, t) = v_n(f, t)V_n(f)R_x^{-1}(f, t)x(f, t) \]
  \[ R_x(f, t) : \text{sum of covariance matrix of all sources} \]
  \[ v_n(f, t) \text{ and } V_n(f) \text{ are estimated by using EM algorithm} \]

• c.f. Beamforming:
  \[ y(f, t) = \sum_{m=0}^{M} W_m(f)x(f, t) \in \mathbb{C} \]
  – Beamforming outputs single-channel signal : MISO
  – MCWF outputs multi-channel signal : MIMO

• How did we utilize multi-channel signal \[ c_n(f, t) \]?
  – 1. Data augmentation
  – 2. Preprocessor of beamforming
Data augmentation using multi-output separation

- All microphone signals from LGM are fed into AM training

Test real (6ch):

13.2%

[Single-mic]

12.2%

[5th mic] [C0 C1 C2] [C0] [LGM single-mic]

10.0%

[LGM data augmentation]

9.2%

[LGM data augmentation + multi-mic]

[Multi-mic]

11.0%

[Fujita et al, Interspeech 2016]
LGM preprocessed beamforming

- $c_n(f, t)$ has 6-ch that holds spatial information \(\rightarrow\) beamforming technique can be applied
- We used cascading of LGM and BeamformIt

[Fujita et al, Interspeech 2016]

Test real (6ch):
- 9.2% $\rightarrow$ 8.8%
LGM preprocessed beamforming

- Update on target source selection
  - Previous system: target source is selected using SRP-PHAT score on the front direction.
  - Sometimes it failed due to permutation errors and how to hold a tablet device.

- Introducing permutation-free modification to LGM
Permutation-free Local Gaussian modeling

• Introducing semi-stationary constraints to noise sources
  
  – Target source \( c_0(f, t) \sim \mathcal{N}_\mathbb{C}(0, v_0(f, t) V_0(f)) \)
  
  – Non-target sources

  \[ c_n(f, t) \sim \mathcal{N}_\mathbb{C}(0, \hat{v}_n(f, t) V_n(f)) \quad (n \geq 1) \]

  – Moving average filter is applied to ‘activity’

  \[ \hat{v}_n(f, t) = \sum_{\tau=0}^{T_n} v_n(v, t - \tau)/T_n \]

  \[ T_1 = 3, \quad T_2 = 6 \]

  – Applying the moving average filter in the each EM iteration, target source, i.e. the most active source is extracted onto \( c_0 \).

  – We no longer select the target source using SRP-PHAT

    Test real(6ch): 8.8% \rightarrow 7.8%
Unsupervised network adaptation

- DNN is self-adapted using 1-best results
- Re-training is performed by using mini-batch SGD with cross entropy criteria [Yoshioka et al, ASRU 2015]
Concerns about re-training of DNN

• Unsupervised adaptation fails when the large number of DNN parameters are adapted [Liao, ICASSP 2012]
  – 32M parameters in our case is medium size. We did not try any parameter reduction technique such as low-rank approximation nor partial layer adaptation
  – Adaptation of entire network works successfully

• Hyper-parameters used in initial training phase is not appropriate for adaptation. We tuned three hyper parameters: learning rate, mini-batch size, and the number of iterations.
  – L2 penalty (weight decay) may be a good option. But we didn’t try it due to time consideration
Unsupervised network adaptation

- Small learning rate, early stopping
- Large mini-batch?

<table>
<thead>
<tr>
<th>iter</th>
<th>Learn rate</th>
<th>mini-batch</th>
<th>dev avg</th>
<th>dev real</th>
<th>dev simu</th>
<th>test real</th>
<th>test simu</th>
</tr>
</thead>
<tbody>
<tr>
<td>No adaptation</td>
<td></td>
<td></td>
<td>4.85</td>
<td>4.49</td>
<td>5.20</td>
<td>7.78</td>
<td>5.20</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>256</td>
<td>4.115</td>
<td>3.93</td>
<td>4.3</td>
<td>6.48</td>
<td>5.05</td>
</tr>
<tr>
<td>1</td>
<td>0.008</td>
<td>512</td>
<td>4.08</td>
<td>3.95</td>
<td>4.21</td>
<td>6.52</td>
<td>4.97</td>
</tr>
<tr>
<td>1</td>
<td>0.001</td>
<td>256</td>
<td>3.7</td>
<td>3.58</td>
<td>3.82</td>
<td>5.56</td>
<td>4.42</td>
</tr>
<tr>
<td>1</td>
<td>0.0004</td>
<td>256</td>
<td>3.745</td>
<td>3.6</td>
<td>3.89</td>
<td>5.66</td>
<td>4.47</td>
</tr>
<tr>
<td>1</td>
<td>0.0004</td>
<td>512</td>
<td>3.735</td>
<td>3.6</td>
<td>3.87</td>
<td>5.67</td>
<td>4.45</td>
</tr>
<tr>
<td>1</td>
<td>0.0004</td>
<td>12000</td>
<td>3.7</td>
<td>3.55</td>
<td>3.85</td>
<td>5.68</td>
<td>4.49</td>
</tr>
<tr>
<td>1</td>
<td>0.0001</td>
<td>256</td>
<td>3.865</td>
<td>3.66</td>
<td>4.07</td>
<td>6.23</td>
<td>4.97</td>
</tr>
<tr>
<td>2</td>
<td>0.0004</td>
<td>12000</td>
<td>3.695</td>
<td>3.58</td>
<td>3.81</td>
<td>5.56</td>
<td>4.47</td>
</tr>
<tr>
<td>10</td>
<td>0.0004</td>
<td>256</td>
<td>4.305</td>
<td>4.1</td>
<td>4.51</td>
<td>6.9</td>
<td>5.4</td>
</tr>
</tbody>
</table>
Phonetically-oriented System Combination

- Combination of 1-best results from various systems
- Word alignment among multiple sentences is important
  - Word based DP matching ⇒ Phonetically-oriented alignment
    [Ruiz et al, ASRU2015]
- Chunk selection using discriminatively trained model

**Multiple Recognizer Outputs**

A  "post and their paying"
B  "cost and there are paying"
C  "cost and they are paying"

**Conventional Word Alignment**

<table>
<thead>
<tr>
<th></th>
<th>post</th>
<th>and</th>
<th></th>
<th>their</th>
<th>paying</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td>there</td>
<td>are</td>
<td>paying</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>they</td>
<td>are</td>
<td>paying</td>
</tr>
</tbody>
</table>

**Selection**

<table>
<thead>
<tr>
<th></th>
<th>cost</th>
<th>and</th>
<th>their</th>
<th>are</th>
<th>paying</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**POWA**

<table>
<thead>
<tr>
<th></th>
<th>post</th>
<th>and</th>
<th>their</th>
<th>paying</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td>there</td>
<td>are</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>they</td>
<td>are</td>
</tr>
</tbody>
</table>

**Selection**

<table>
<thead>
<tr>
<th></th>
<th>cost</th>
<th>and</th>
<th>their</th>
<th>paying</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

POWA: Phonetically-Oriented Word Alignment
Phonetically-oriented System Combination

**Feature vector**

\[ x = (x_{cf}^\top, x_{oc}^\top, x_{nl}^\top)^\top \]

Geometric mean of confidence score in a chunk

\[ x_{cf} = \left( \prod_{e=0}^{E} c_e \right)^{1/E} ; 0 < i < H \]

Co-occerance: whether two chunks are identical

\[ x_{oc} = (\delta(w_i, w_j) ; 0 < i < j < H)^\top \]

NULL: whether a chunk is NULL

\[ x_{nl} = (\delta(w_i, \text{NULL}) ; 0 < i < H)^\top \]

**Label**

\[ y = (\delta(w_h, w_{true}) ; 0 < h < H)^\top \]

Logistic regression model are trained using development set
Phonetically-oriented System Combination

• 12 backend models
  – 4 baselines {GMM, DNN+sMBR, DNN+5-gram, RNNLM}
  – 4 data augmented models
  – 2 adapted DNN models {5-gram, RNNLM}
  – 2 data augmented + adapted DNN models

• 2 frontend speech enhancement
  – Baseline (beamformit)
  – LGM preprocessed beamforming

Test real WERs on 6ch track

<table>
<thead>
<tr>
<th>Best single recognizer</th>
<th>5.56 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>24-recognizer combination (conventional)</td>
<td>4.75 %</td>
</tr>
<tr>
<td>24-recognizer combination with POWA</td>
<td>4.68 %</td>
</tr>
</tbody>
</table>
Summary of experimental evaluation

- LGM based system reduce WER especially on 6ch track
- Speaker adaptation is effective when base WER is low
- System combination reduce WER on all tracks.

<table>
<thead>
<tr>
<th>system</th>
<th>Test real WER(%)</th>
<th>Rel. improvement(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1ch</td>
<td>2ch</td>
</tr>
<tr>
<td>Baseline</td>
<td>23.59</td>
<td>16.6</td>
</tr>
<tr>
<td>LGM(data augmented + beamforming)</td>
<td>16.88</td>
<td>12.1</td>
</tr>
<tr>
<td>LGM+ speaker adaptation</td>
<td>13.57</td>
<td>9.09</td>
</tr>
<tr>
<td>system combination</td>
<td>11.42</td>
<td>8.61</td>
</tr>
</tbody>
</table>
Conclusion

• Local Gaussian modeling based source separation
  – Multi-channel Wiener filter output is useful
  – Introducing semi-stationary constraint to non-target sources improves frontend speech enhancement
  – Achieved up to 32.1% gain from baseline

• Unsupervised deep neural network adaptation
  – Unsupervised re-training of DNN works well for speaker adaptation when using conservative training parameters
  – Achieved up to 28.5% gain on 6ch track

• Phonetically-oriented system combination
  – Word alignment considering phonetic similarity improves system combination
  – Achieved up to 15.8% gain on 6ch track
Future work

- Cross-adaptation or Committee-based approach [Kanda et al, Interspeech2016] for speaker adaptation
  - Supervision from other systems gives better performance
- Noise environment adaptation
  - Noise adaptation is more desired than speaker adaptation in robot applications; a speaker in front of a robot changes rapidly but noise environment is relatively fixed
- Deep learning based multi-channel Wiener filter and joint training of the filter and acoustic model
  - Many studies on this field are found in Interspeech 2016
HITACHI
Inspire the Next