The JHU Multi-Microphone Multi-Speaker ASR System for the CHiME-6 Challenge

Challenge Overview

Track 1
- ASR only
- Oracle speaker segments provided

Track 2
- Diarization + ASR
- No speaker segments
Challenge Overview

Track 1
- ASR only
- Oracle speaker segments provided

Track 2
- Diarization + ASR
- No speaker segments

We will present our Track-2 system.

Same ASR model used in both tracks.
Our Track-2 Pipeline

- CH1
- CH2
- CH3
- CH4

- U01
- U06

- WPE
- Beamforming

- SAD with multi-array posterior fusion

- Diarization with multi-array PLDA score fusion

- VB-based overlap assignment

- Multi-array GSS

- ASR
  - Acoustic Model
  - Language Model
  - Lattice Combination

- Transcription

Replaced oracle in track 1
Speech Enhancement

- SAD with multi-array posterior fusion
- Diarization with multi-array PLDA score fusion
- VB-based overlap assignment
- Multi-array GSS
- ASR
  - Acoustic Model
  - Language Model
  - Lattice Combination
- Transcription

Replaced oracle in track 1
Speech Enhancement

WPE & BeamformIt

- The first preprocessing step performed was weighted prediction error (WPE*) based online **multi-channel** dereverberation^.

- Subsequently a delay and sum beamformer (BeamformIt+) was used to denoise the signal.

---


Denoising Alternative

- Trained a neural network to separate noise from speech mixtures.
- Used voxceleb data for simulation with CHiME-6 noises (mixtures of 1-4 speakers).
- Perceptually seemed better in very noisy segments but didn’t help final performance. So it was not used in our final system.
Other Ideas We Tried

BSS & TaSNet

• If blind speech separation (BSS) can be performed before speaker segmentation we won’t require speaker diarization.

• We tried two methods: (1) independent vector analysis (IVA)* and (2) TaSNet†

• We were not able to obtain good results with this approach, so it was not used in our final system.

Guided source separation (GSS) was used to separate the target source using the time annotations.

The groundtruth annotations were used for Track 1 and diarization outputs were used to obtain the time annotations for Track 2.

Importance of Overlap Assignment for GSS – Track 2

Multi-Array GSS - Sensitivity to Diarization Output

<table>
<thead>
<tr>
<th>Method</th>
<th>Overlap Detection</th>
<th>Dev WER (%)</th>
<th>Eval WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Array GSS</td>
<td>N</td>
<td>71.0</td>
<td>68.8</td>
</tr>
<tr>
<td>Multi-Array GSS</td>
<td>Y</td>
<td>69.3</td>
<td>68.8</td>
</tr>
</tbody>
</table>

- GSS is very sensitive to the diarization output.
- VB-HMM based overlap detection helps GSS
Combining Early & Late Fusion – Track 2

Multi-View Decoding with GSS

- Early fusion was performed by incorporating all arrays while beamforming.
- Late fusion was performed by lattice combination of GSS output on multi-array with individual arrays.

<table>
<thead>
<tr>
<th>Separation Method</th>
<th>Early Fusion</th>
<th>Late Fusion</th>
<th>Dev WER (%)</th>
<th>Eval WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Array GSS</td>
<td>Y</td>
<td>N</td>
<td>69.3</td>
<td>68.8</td>
</tr>
<tr>
<td>Single-Array GSS (U06)</td>
<td>N</td>
<td>N</td>
<td>73.1</td>
<td>72.1</td>
</tr>
<tr>
<td>Single-Array GSS (U04)</td>
<td>N</td>
<td>N</td>
<td>72.3</td>
<td>74.4</td>
</tr>
<tr>
<td>Multi-Array GSS + Single-Array GSS</td>
<td>Y</td>
<td>Y</td>
<td>68.3</td>
<td>68.3</td>
</tr>
</tbody>
</table>
Combining Early & Late Fusion – Track 2

Multi-View Decoding with GSS

- Early fusion was performed by incorporating all arrays while beamforming.
- Late fusion was performed by lattice combination of GSS output on multi-array with individual arrays.

<table>
<thead>
<tr>
<th>Separation Method</th>
<th>Early Fusion</th>
<th>Late Fusion</th>
<th>Dev WER (%)</th>
<th>Eval WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Array GSS</td>
<td>Y</td>
<td>N</td>
<td>69.3</td>
<td>68.8</td>
</tr>
<tr>
<td>Single-Array GSS (U06)</td>
<td>N</td>
<td>N</td>
<td>73.1</td>
<td>72.1</td>
</tr>
<tr>
<td>Single-Array GSS (U04)</td>
<td>N</td>
<td>N</td>
<td>72.3</td>
<td>74.4</td>
</tr>
<tr>
<td>Multi-Array GSS + Single-Array GSS</td>
<td>Y</td>
<td>Y</td>
<td>68.3</td>
<td>68.3</td>
</tr>
</tbody>
</table>
Combining Early & Late Fusion – Track 2

Multi-View Decoding with GSS

<table>
<thead>
<tr>
<th>Separation Method</th>
<th>Early Fusion</th>
<th>Late Fusion</th>
<th>Dev WER (%)</th>
<th>Eval WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Array GSS</td>
<td>Y</td>
<td>N</td>
<td>69.3</td>
<td>68.8</td>
</tr>
<tr>
<td>Single-Array GSS (U06)</td>
<td>N</td>
<td>N</td>
<td>73.1</td>
<td>72.1</td>
</tr>
<tr>
<td>Single-Array GSS (U04)</td>
<td>N</td>
<td>N</td>
<td>72.3</td>
<td>74.4</td>
</tr>
<tr>
<td>Multi-Array GSS + Single-Array GSS</td>
<td>Y</td>
<td>Y</td>
<td>68.3</td>
<td>68.3</td>
</tr>
</tbody>
</table>

- Early fusion was performed by incorporating all arrays while beamforming.
- Late fusion was performed by lattice combination of GSS output on multi-array with individual arrays.
Combining Early & Late Fusion – Track 2

Multi-View Decoding with GSS

Early Fusion

- Multi-Array GSS
- Single-Array GSS (U06)
- Single-Array GSS (U04)
- Multi-Array GSS + Single-Array GSS

Late Fusion

Lattice Weights

<table>
<thead>
<tr>
<th>Separation Method</th>
<th>Early Fusion</th>
<th>Late Fusion</th>
<th>Dev WER (%)</th>
<th>Eval WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Array GSS</td>
<td>Y</td>
<td>N</td>
<td>69.3</td>
<td>68.8</td>
</tr>
<tr>
<td>Single-Array GSS (U06)</td>
<td>N</td>
<td>N</td>
<td>73.1</td>
<td>72.1</td>
</tr>
<tr>
<td>Single-Array GSS (U04)</td>
<td>N</td>
<td>N</td>
<td>72.3</td>
<td>74.4</td>
</tr>
<tr>
<td>Multi-Array GSS + Single-Array GSS</td>
<td>Y</td>
<td>Y</td>
<td>68.3</td>
<td>68.3</td>
</tr>
</tbody>
</table>

- Early fusion was performed by incorporating all arrays while beamforming.
- Late fusion was performed by lattice combination of GSS output on multi-array with individual arrays.
Speech Enhancement Takeaways

GSS gives good improvements and it is sensitive to the diarization output

Further improvement can be obtained by combining early and late fusions
Speech Activity Detection
Speech Activity Detection

- We built a TDNN-Stats based architecture for baseline
- We use the same core model
- NEW: Multi-array extension

$P(s|x)$

Softmax

TDNN

Stats pooling

TDNN

Stats pooling

TDNN

TDNN

TDNN

$\mathbf{x}_{i-k}, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_{i+k}$
Baseline SAD

Beamforming over channels within array

\[ P(s|x) \rightarrow \text{segments} \]
Baseline SAD

Random selection of array U06

U01
CH1
CH2
CH3
CH4
WPE
Beamforming
TDNN-Stats neural network
$P(s|x)$
segments

U06
CH1
CH2
CH3
CH4
WPE
Beamforming
TDNN-Stats neural network
$P(s|x)$
segments
U06 is not the best selection
Multi-channel + multi-array

1. Can we do better than beamforming for channel combination?

2. Can we do better than random array selection?
Can channel-level posterior fusion do better than beamforming?

Diagram:
- CH1, CH2, CH3, CH4 connected to WPE
- WPE connected to TDNN-Stats
- TDNN-Stats connected to $P(s | x)$
- $P(s | x)$ connected to Posterior fusion
- Posterior fusion connected to segments

Diary with multi-array PLDA score fusion
VB-based overlap assignment
Multi-array GSS

ASR
- Acoustic Model
- Language Model
- Lattice Combination

Transcription
Posterior Mean < Posterior Max ≈ Beamforming

But 4x more compute -> keep Beamforming!
Posterior fusion for array combination
Posterior fusion helps array combination

SAD with multi-array posterior fusion
Diarization with multi-array PLDA score fusion
VB-based overlap assignment
Multi-array GSS

ASR
Acoustic Model
Language Model
Lattice Combination
Transcription

Error %
0 2 4 6 8

Array
U01 U02 U03 U04 U06
Posterior avg Posterior max

Missed speech False alarm

5.3 5.1 4.1 6.6 5.4
2.2 2.1 2.1 2.1 2 2.1 2.1 2.1 2.2
3.4 4.6

TDNN-SWDWV QHXUDO QHWZRUN
Posterior fusion helps array combination significantly

- Huge improvement in missed speech
- Directly affects downstream DER and WER

![Graph showing error rates for different arrays with posterior fusion](image)

- Missed speech
- False alarm

Error %

- U01: 2.2
- U02: 2.1
- U03: 2.1
- U04: 2.1
- U06: 4.6
- Posterior avg: 3.4
- Posterior max: 5.5

34.3% decrease in missed speech
SAD Takeaways

Beamforming is better than channel-level posterior fusion.

Array-level posterior fusion gives 34% improvement.
Speaker Diarization with multi-array PLDA score fusion

SAD with multi-array posterior fusion

VB-based overlap assignment

Multi-array GSS

ASR

Acoustic Model

Language Model

Lattice Combination

Transcription
Baseline system

Baseline SAD

segments

Beamforming

x-vector extraction 1.5s/0.75s

PLDA scoring

AHC

RTTM

SAD with multi-array posterior fusion

Diarization with multi-array PLDA score fusion

VB-based overlap assignment

Multi-array GSS

ASR

Acoustic Model

Language Model

Lattice Combination

Transcription
Better SAD improves diarization

DER on dev for U06 (evaluated using original reference without UEM)

63.49% -> 58.95%
(59.82% mean DER of all arrays)

All further results using multi-array SAD
Two Questions

1. Only 1 array is used -> how to use **multi-array** information?

2. Overlaps are ignored -> how to handle **overlapping speakers**?
**DOVER to combine array outputs**

**Table: System Performance**

<table>
<thead>
<tr>
<th>System</th>
<th>Dev DER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (mean)</td>
<td>59.8</td>
</tr>
<tr>
<td>DOVER</td>
<td>61.1</td>
</tr>
</tbody>
</table>

PLDA score fusion helps slightly

### System Dev DER (%)

<table>
<thead>
<tr>
<th></th>
<th>Dev DER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (mean)</td>
<td>59.8</td>
</tr>
<tr>
<td>PLDA score MEAN</td>
<td>59.9</td>
</tr>
<tr>
<td>PLDA score MAX</td>
<td>59.0</td>
</tr>
</tbody>
</table>
Can we do better?

- Small improvement in DER, but underwhelming!
- Maybe need to break segments into more pieces?
Use 0.25s shift for x-vector extraction

Better DER with smaller segments

<table>
<thead>
<tr>
<th>System</th>
<th>Dev DER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (mean)</td>
<td>59.8</td>
</tr>
<tr>
<td>PLDA score MAX</td>
<td>59.0</td>
</tr>
<tr>
<td>+ 0.25s shift</td>
<td>57.9</td>
</tr>
</tbody>
</table>

**System Diagram**

- **Multi-array SAD**
- **Segments**
- **Beamforming**
  - **x-vector extraction 1.5s/0.25s**
  - **PLDA scoring**
  - **PLDA Score Fusion**
  - **AHC**
  - **RTTM**

**ASR**
- **Acoustic Model**
- **Language Model**
- **Lattice Combination**
- **Transcription**
Overlap Handling

Overlap Handling

Overlap assignment:
1. Heuristic-based
2. VB-HMM based

Simple heuristic gives small DER improvement

<table>
<thead>
<tr>
<th>System</th>
<th>Dev DER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (mean)</td>
<td>59.8</td>
</tr>
<tr>
<td>PLDA score MAX</td>
<td>59.0</td>
</tr>
<tr>
<td>+ 0.25s shift</td>
<td>57.9</td>
</tr>
<tr>
<td>Overlap (heuristic)</td>
<td>56.2</td>
</tr>
</tbody>
</table>
VB-HMM gives large DER improvement

<table>
<thead>
<tr>
<th>System</th>
<th>Dev DER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (mean)</td>
<td>59.8</td>
</tr>
<tr>
<td>PLDA score MAX</td>
<td>59.0</td>
</tr>
<tr>
<td>+ 0.25s shift</td>
<td>57.9</td>
</tr>
<tr>
<td>Overlap (heuristic)</td>
<td>56.2</td>
</tr>
<tr>
<td>Overlap (VB-HMM)</td>
<td><strong>50.4</strong></td>
</tr>
</tbody>
</table>
Diarization Takeaways

Multi-array PLDA fusion helps with smaller segments.

VB-HMM overlap assignment helps significantly!
• HMM-GMM/DNN with LFMMI Objective
  • Training data: combination of worn mic utterances (80h), beamformed array data (160h), and multi-array GSS enhanced data (40h)
  • Pronunciation and Silence modeling is used in HMM-GMM training stage.
  • Speed perturbation is used in LFMMI training stage.
Optimizing Model Architecture

- CNN-TDNN-F model outperforms other models on CHiME 6 data.
- Oracle overlap information bit is appended to the Fbank features.

<table>
<thead>
<tr>
<th>Model Architectures</th>
<th>Dev WER (%)</th>
<th>Eval WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDNN-F</td>
<td>51.8</td>
<td>51.3</td>
</tr>
<tr>
<td>CNN-TDNN-LSTM</td>
<td>50.1</td>
<td>49.8</td>
</tr>
<tr>
<td>SA-CNN-TDNN-F</td>
<td>49.9</td>
<td>49.4</td>
</tr>
<tr>
<td>CNN-TDNN-F</td>
<td>48.3</td>
<td>48.5</td>
</tr>
</tbody>
</table>
Training Data Selection

- Enhanced far-field data outperforms raw far-field and simulated data.
- Data selection speeds up experiments.
- Speed perturbation in all cases.

<table>
<thead>
<tr>
<th>Training data</th>
<th>Dev WER (%)</th>
<th>Eval WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worn Mic (80h)</td>
<td>49.6</td>
<td>49.4</td>
</tr>
<tr>
<td>Worn Mic + Aug (320h)</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Array (200h)</td>
<td>44.6</td>
<td>45.4</td>
</tr>
<tr>
<td>Array + Beamformit (160h)</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Array + GSS (40h)</td>
<td>44.5</td>
<td>44.9</td>
</tr>
</tbody>
</table>

Appending Oracle Overlap Information (Track 1)

- Using overlap information as auxiliary input for AM training obtains slight WER improvements.

- Track 2 experiments in progress.

<table>
<thead>
<tr>
<th>Overlap info in ivector</th>
<th>Overlap info in nnet</th>
<th>Dev WER</th>
<th>Eval WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>no</td>
<td>44.5</td>
<td>44.9</td>
</tr>
<tr>
<td>yes</td>
<td>no</td>
<td>44.9</td>
<td>45.2</td>
</tr>
<tr>
<td>yes</td>
<td>yes</td>
<td>44.3</td>
<td>44.4</td>
</tr>
</tbody>
</table>
Neural LM and Rescoring

- A forward and a backward LSTM
  - Each is a 2-layer projected LSTM
  - Hidden dim = 512, projection dim = 128
  - Backward LSTM is trained on transcription reversed on sentence level

- 2-stage pruned lattice rescoring
  - Stage 1: Forward LSTM
  - Stage 2: Backward LSTM

- Kaldi for neural LM training and rescoring
Results and Discussion
Step-by-Step Improvements for Track 1

- Baseline: 51.75
- CNN-TDNN-F AM: 51.28
- + Augmentation: 49.59
- + Overlap Feature: 49.37
- + LM Rescoring: 44.37
- + Lattice Combination: 42.82
- + Lattice Combination and Rescoring: 41.75

Improvements:
- 22.2% lower
- 21.1% lower
Step-by-Step Improvements for Track 2
Summary

Frontend
- GSS performance improved with improvement in DER.

SAD
- Multi-array posterior fusion improves error rate by 34% relative

Diarization
- Multi-array PLDA score fusion shows small improvement in DER
- VB-HMM based overlap assignment shows large DER gain and some WER improvement

Acoustic Model
- Deep CNN-TDNN-F model is an effective architecture.
- Enhanced far-field data outperforms raw far-field and simulated data as data augmentation.

Language model
- Neural LM rescoring obtains modest WER reductions.