End-to-End Multi-speaker Speech Recognition with Transformer

Xuankai Chang\textsuperscript{1}, Wangyou Zhang\textsuperscript{2},
Yanmin Qian\textsuperscript{2}, Jonathan LeRoux\textsuperscript{3}, Shinji Watanabe\textsuperscript{1}
\textsuperscript{1}Center for Language and Speech Processing, Johns Hopkins University, USA
\textsuperscript{2}SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
\textsuperscript{3}Mitsubishi Electric Research Laboratories (MERL), USA
• Multi-speaker speech processing (Cocktail party problem)
End-to-End is attractive

✓ No need for parallel clean audios
End-to-End is attractive

- No need for parallel clean audios
- Simplifying the complicated model-building
End-to-End is attractive

- No need for parallel clean audios
- Simplifying the complicated model-building
- Natural incorporation with Linguistic Information
End-to-End speech recognition

**Single-input, Single-output**

Neural Network  
[Graves+ 2014]

“ICASSP is interesting.”

**Multi-input, Single-output**

Neural Network  
[Ochiai+ 2017]

“ICASSP is interesting.”
End-to-End speech recognition

Single-input, Single-output

Neural Network
[Graves+ 2014]

“ICASSP is interesting.”

Multi-input, Single-output

Neural Network
[Ochiai+ 2017]

“ICASSP is interesting.”
End-to-End speech recognition

Single-input, Multi-output

Neural Network
[Seci+ 2018]

“ICASSP is interesting.”

“Virtual conference is fresh.”

Multi-input, Multi-output

Neural Network
[Chang+ 2019]

“ICASSP is interesting.”

“Virtual conference is fresh.”
End-to-End speech recognition

Single-input, Multi-output
Neural Network
[Seki+ 2018]

“ICASSP is interesting.”
“Virtual conference is fresh.”

Multi-input, Multi-output
Neural Network
[Chang+ 2019]

“ICASSP is interesting.”
“Virtual conference is fresh.”
End-to-End speech recognition

Single-input, Multi-output

Bi-LSTM

“ICASSP is interesting.”

“Virtual conference is fresh.”

Multi-input, Multi-output

Bi-LSTM

“ICASSP is interesting.”

“Virtual conference is fresh.”
End-to-End speech recognition

Transformer

Single-input, Multi-output

Transformer

“ICASSP is interesting.”

“Virtual conference is fresh.”

Multi-input, Multi-output

Transformer

“ICASSP is interesting.”

“Virtual conference is fresh.”
Single-Channel Multi-Speaker E2E ASR

1. **Encoder**: separating and encoding as high dimensional representation
2. **Decoder**: generating the output token sequence
3. **CTC**: determining the permutation of reference sequences

Hiroshi Seki, et al. “A purely end-to-end system for multi-speaker speech recognition”, ACL, 2018
Single-Channel Multi-Speaker E2E ASR

1. **Encoder**: separating and encoding as high dimensional representation
2. **Decoder**: generating the output token sequence
3. **CTC**: determining the permutation of reference sequences

Hiroshi Seki, et al. “A purely end-to-end system for multi-speaker speech recognition”, ACL, 2018
1. **Speech separation**: Multi-source mask-based neural beamformer
2. **Feature extraction**: STFT $\rightarrow$ Log Mel-filterbank
3. **Speech recognition**: Joint CTC/attention-based encoder-decoder

Multi-Channel Multi-Speaker ASR

1. Speech separation: Multi-source mask-based neural beamformer
2. Feature extraction: STFT → Log Mel-filterbank
3. Speech recognition: Joint CTC/attention-based encoder-decoder

Transformer (Self-attention)

- Self-attention

  ![Diagram of Transformer (Self-attention)]

  - Linear Projection (Q)
  - Linear Projection (K)
  - Linear Projection (V)
  - Weighted Summation
  - Attention Output
  - Queries
  - Keys
  - Values
  - Hidden State
Transformer (Self-attention)

- Self-attention

Encoder

CTC

Attention Decoder

Self-attention

Hidden State

Linear Projection (Q)

Linear Projection (K)

Linear Projection (V)

Weighted Summation

Attention Output

Queries

Keys

Values
Transformer (Self-attention)

- Time-restricted Self-attention
Transformer (Self-attention)

- Time-restricted Self-attention

Self-attention

Linear Projection (Q)

Linear Projection (K)

Linear Projection (V)

Hidden State

Attention Output

Weighted Summation

Masking Network

Beam-forming

\{X\}^C_{c=1}

Queries

Keys

Values

Sliding window (30 frames: 14 on the left, 15 on the right)
## Experiment – Data

<table>
<thead>
<tr>
<th>Data</th>
<th>Name</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-channel single-speaker</td>
<td>WSJ</td>
<td>-</td>
</tr>
<tr>
<td>Single-channel multi-speaker</td>
<td>wsj1-2mix [1]</td>
<td>-</td>
</tr>
<tr>
<td>Multi-channel multi-speaker</td>
<td>Spatialized wsj1-2mix¹</td>
<td>Train: 98.5 hr</td>
</tr>
<tr>
<td></td>
<td>2 versions:</td>
<td>Dev: 1.3 hr</td>
</tr>
<tr>
<td></td>
<td>• Anechoic</td>
<td>Eval: 0.8 hr</td>
</tr>
<tr>
<td></td>
<td>• Reverberant</td>
<td></td>
</tr>
</tbody>
</table>

¹ The spatialization toolkit is available at http://www.merl.com/demos/deep-clustering/spatialize_wsj0-mix.zip

Results – Single-channel multi-speaker

- Anechoic
  - 1st Channel

- Reverberant
  - Nara-WPE preprocessing
  - 1st Channel

![Graph showing WER for Anechoic and Reverberated conditions with 1st Channel results.](image)
Results – Single-channel multi-speaker

- Anechoic
  - 1st Channel

- Reverberant
  - Nara-WPE preprocessing
  - 1st Channel
• Anechoic
  • 1st Channel
• Reverberant
  • Nara-WPE preprocessing
  • 1st Channel
1. Include original WSJ (single-channel single speaker)
   • Bypassing the frontend
   • Helps regularize training
     • Improves backend ASR performance
     • Benefits frontend performance

2. Curriculum Learning
   • In the order of balanced $\rightarrow$ unbalanced energy between the sources
     1) balanced means both streams in the frontend can be trained.
     2) unbalanced samples to refine one of the streams.
Results – Multi-channel multi-speaker

- **Anechoic**
  - Dev WER: 8.6, 10.7, 13.5
  - Eval WER: 6.9, 6.4

- **Reverberant**
  - Dev WER: 30.0, 33.0, 35.0
  - Eval WER: 28.0, 26.0
Results – Multi-channel multi-speaker

• Anechoic

- Eval WER: RNN-Frontend + RNN-Backend: 11.8, RNN-Frontend + Transformer-Backend: 6.9, Transformer-Frontend + Transformer-Backend: 6.4

• Reverberant

- Dev WER: RNN-Frontend + RNN-Backend: 35.0, RNN-Frontend + Transformer-Backend: 31.9, Transformer-Frontend + Transformer-Backend: 28.0
- Eval WER: RNN-Frontend + RNN-Backend: 33.0, RNN-Frontend + Transformer-Backend: 30.0, Transformer-Frontend + Transformer-Backend: 26.0

Anechoic setting: 13.2% WER decrease compared to Reverberant setting.
Results – Multi-channel

With external dereverberation (WPE)

<table>
<thead>
<tr>
<th></th>
<th>Dev (Dereverberated)</th>
<th>Eval (Dereverberated)</th>
<th>Dev (Reverberant)</th>
<th>Eval (Reverberant)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN-Frontend + RNN-Backend</td>
<td>35.0</td>
<td>33.0</td>
<td>31.9</td>
<td></td>
</tr>
<tr>
<td>RNN-Frontend + Transformer-Backend</td>
<td>30.0</td>
<td>28.0</td>
<td>26.0</td>
<td></td>
</tr>
<tr>
<td>Transformer-Frontend + Transformer-Backend</td>
<td>24.5</td>
<td>19.2</td>
<td>20.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17.7</td>
<td>15.2</td>
<td>15.5</td>
<td></td>
</tr>
</tbody>
</table>
Results – Multi-channel

- With external dereverberation (WPE)

![Graph showing results with and without dereverberation for different models.](image)
Speech Separation Ability

Multi-channel multi-speaker end-to-end ASR

Speech separation and enhancement

Feature extractor

Speech recognition

Overlapped Segment

Separated Segment 1

Separated Segment 2

Masking Network

Beamforming

X

Log Mel-filterbank

Log Mel-filterbank

Encoder

Encoder

CTC

CTC

$L_{ctc}$

$L_{att}$
Conclusion

- Transformer based multi-speaker end-to-end ASR
  - Single-channel
  - Multi-channel
    - Backend ASR: encoder & decoder
    - Frontend masking network: local self-attention
      - First to apply self-attention in speech separation.

- Future work
  - To improve the performance of the model with Transformer frontend
  - To integrate dereverberation in the system
  - To apply the model on real data
Thanks!

Q & A

• Special thanks to my co-authors:

Wangyou Zhang

Yanmin Qian

Jonathan Le Roux

Shinji Watanabe