CS11-737 Multilingual NLP
Streaming Speech Translation
Lei Li
https://lileicc.github.io/course/11737mnlp23fa/
Simultaneous Speech-to-text Translation

- Read the audio signals of speech in one language, and translate to the text in another language while speaker speaks (SiST).
Wide Applications of SST

Foreign Media

Global Conferences

Tourism

International Trade
- **Drawbacks:**
  1. Computationally inefficient
  2. Error propagation:
     Wrong/error transcript recognition $\rightarrow$ Wrong translation
- **Goal**: End2end streaming ST needs to balance the latency and quality, and generate translations based on the partial speech chunk with a single model.

- **Predecessor’s method**: Wait-K
Challenges for SiST

- **Latency**: Low latency is required for better user experience. → Translate as early as possible.

- **Applicability**: ...

- **Accuracy**: More context is required to improve speech translation. → Wait as long as possible.

- **Flicker**: ...

- **...**
Simple Approach: Wait-K with Fixed Stride

① Listen to streaming speech with **a fixed stride** after \( K \) steps.

② Do **listen** and **write** iteratively till the end.

\( \text{Ich erinnere mich an mein erstes Feuer.} \)
MoSST: Key Insight

Motivation: How to find *proper moments* to generate partial sentence translation given a streaming speech input?

Solution: Introduce a *monotonic segmentation module*.

![Diagram showing the workflow of Acoustic Encoder, Monotonic Segmentation Module, and Semantic Encoder, with supervision on Length of Transcription.](image_url)
MoSST Overview

Audio Waveform

Acoustic Encoder

Monotonic Segmentation Module

Self Attention

Feed Forward

x N

LP loss

CE loss

x N

Feed Forward

Cross Attention

Self Attention

Translation

Transcription
Monotonic Segmentation while Listening

(source: Dong et al., 2020[2])

MoSST: Training Strategies

• Full-sentence training without Wait-K is ok!

• To alleviate the data scarcity problem:
  ○ Pre-trained Acoustic Model
  ○ Multi-task Training
MoSST Adaptively decide when to Generate Translation

- Adaptive Decision vs Pre-fixed Decision

\[ |x - y| < K \]

\[ \text{read} \]

\[ \text{write} \]

\[ \text{exit?} \]

Listen

Write

\[ \text{Ich erinnere mich an mein erstes Feuer.} \]

\[ \text{I remember my first fire.} \]
Experimental Setups

1. Datasets
MuST-C, En\(\mathbb{D}\) De/Fr

2. Metrics
Accuracy
- BLEU
Latency
- Differentiable Average Lagging
- Average Proportion
- Average Lagging

3. Model

<table>
<thead>
<tr>
<th>Module</th>
<th>Backbone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acoustic</td>
<td>Wav2vec 2</td>
</tr>
<tr>
<td>Encoder</td>
<td>MSM</td>
</tr>
<tr>
<td>Semantic</td>
<td>Transformer</td>
</tr>
<tr>
<td>Encoder</td>
<td>Encoder</td>
</tr>
<tr>
<td>Decoder</td>
<td>Transformer</td>
</tr>
<tr>
<td></td>
<td>Decoder</td>
</tr>
</tbody>
</table>
MoSST works much better

MoSST achieves best translation accuracy with the same lagging.
MoSST Is Better Than SimulST (Ma et al., 2020b)

MoSST achieves best translation accuracy with the same lagging.
MoSST Is Better Than SimulSpeech (Ren et al., 2020)
MoSST Is Superior to Cascaded SiST

```
<table>
<thead>
<tr>
<th>k</th>
<th>BLEU</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.24</td>
<td>32.8</td>
</tr>
<tr>
<td>2</td>
<td>11.24</td>
<td>32.8</td>
</tr>
<tr>
<td>3</td>
<td>11.24</td>
<td>32.8</td>
</tr>
<tr>
<td>4</td>
<td>11.24</td>
<td>32.8</td>
</tr>
<tr>
<td>5</td>
<td>11.24</td>
<td>32.8</td>
</tr>
<tr>
<td>6</td>
<td>11.24</td>
<td>32.8</td>
</tr>
<tr>
<td>7</td>
<td>11.24</td>
<td>32.8</td>
</tr>
<tr>
<td>∞</td>
<td>11.24</td>
<td>32.8</td>
</tr>
</tbody>
</table>
```

- **BLEU**: A measure of the quality of a translation, higher is better.
- **WER**: Word Error Rate, lower is better.

MoSST consistently outperforms Cascaded SiST across all values of k.
MoSST also works for Offline ST

<table>
<thead>
<tr>
<th>Model</th>
<th>EN-&gt;DE</th>
<th>EN-&gt;FR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer ST Fairseq (Wang et al., 2020)</td>
<td>22.7</td>
<td>32.9</td>
</tr>
<tr>
<td>Transformer ST ESPnet (Inaguma et al., 2020)</td>
<td>22.9</td>
<td>32.8</td>
</tr>
<tr>
<td>Transformer ST NeurST (Zhao et al., 2021)</td>
<td>22.8</td>
<td>33.3</td>
</tr>
<tr>
<td>AFS ST (Zhang et al., 2020)</td>
<td>22.4</td>
<td>31.6</td>
</tr>
<tr>
<td>STAST (Liu et al., 2020)</td>
<td>23.1</td>
<td>-</td>
</tr>
<tr>
<td>Dual-Decoder Transformer (BL) (Le et al., 2020)</td>
<td>23.6</td>
<td>33.5</td>
</tr>
<tr>
<td>Wav2Vec2 + Transformer (Han et al., 2021)</td>
<td>22.3</td>
<td>34.3</td>
</tr>
<tr>
<td>W-Transf (Ye et al., 2021)</td>
<td>23.6</td>
<td>34.6</td>
</tr>
<tr>
<td>RealTrans (Zhang et al., 2021)</td>
<td>22.99</td>
<td>-</td>
</tr>
<tr>
<td><strong>MoSST</strong></td>
<td><strong>24.9</strong></td>
<td><strong>35.3</strong></td>
</tr>
</tbody>
</table>

(under the constrained setting)
Simultaneous Training (prefix-to-prefix) Is Not Necessary for SiST

- ConvTransformer with offline ASR pre-training

**Graph:**

<table>
<thead>
<tr>
<th>BLEU</th>
<th>train-k test-k</th>
<th>train-full test-k</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Differentiable Average Lagging

*Offline training is better than online training with Wait-K.*
Fixed Strides: Bigger Stride, Higher Latency

BLEU

Differentiable Average Lagging

Stride=280ms
Stride=320ms
Stride=400ms
Stride=480ms
Adaptive Decision Is Better Than Pre-fix Decision

![Graph showing the comparison between pre-fix and adaptive decisions. The graph plots BLEU scores against Differentiable Average Lagging. Pre-fix decisions show a lower BLEU score than adaptive decisions across all values of Differentiable Average Lagging.](image-url)
Monotonic Segmentation (MSM) is Important! 

<table>
<thead>
<tr>
<th></th>
<th>MoSST</th>
<th>w/o MSM</th>
<th>w/o MTL</th>
<th>w/o Pretrain</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN→DE</td>
<td>24.9</td>
<td>22.7</td>
<td>21.9</td>
<td>20</td>
</tr>
<tr>
<td>EN→FR</td>
<td>35.3</td>
<td>34.4</td>
<td>33.8</td>
<td>31.6</td>
</tr>
</tbody>
</table>

BLEU
## Case Study

### Source Speech

<table>
<thead>
<tr>
<th></th>
<th>En (Source)</th>
<th>De (Target)</th>
<th>ASR</th>
<th>Cascades</th>
<th>MoSST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>If you have something to give , give it now .</td>
<td>Wenn Sie etwas zu geben haben , geben Sie es jetzt .</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>If you have something to give and give it now .</td>
<td>Wenn Sie etwas zu geben haben und es jetzt geben .</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Wenn Sie etwas geben , geben Sie es jetzt .</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Pause in source speech matters!**
• End-to-end SiST is a more challenging area that requires balancing accuracy and latency.

• To segment audio waveform into acoustic units, MoSST introduces a new monotonic segmentation module, based on which the adaptive decision strategy can dynamically decide when to translate in streaming scenarios.

• MoSST can significantly outperform SOTA baselines both for streaming and non-streaming ST.
Language in 10