ICT’s System for AutoSimTrans 2021: Robust Char-Level Simultaneous Translation

Shaolei Zhang\textsuperscript{1,2}, Yang Feng\textsuperscript{1,2}\ast

\textsuperscript{1}Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
\textsuperscript{2}University of Chinese Academy of Sciences, Beijing, China
{zhangshaolei20z, fengyang}@ict.ac.cn
Contents

Motivation

Method

Experiments

Conclusion
Pipeline of simultaneous interpretation
- Automatic Speech Recognition (ASR) → simultaneous translation (ST) → Text-to-Speech Synthesis (TTS)

Input of simultaneous translation:
- Inaccurate, unsegmented.
- Spoken language domain.

Robustness and Domain adaptability

<table>
<thead>
<tr>
<th>Streaming Transcript</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>大家好欢迎大家来来这里</td>
<td>Hello everyone! Welcome everyone come here.</td>
</tr>
</tbody>
</table>
### Motivation

#### For robustness
- ASR result (streaming transcription): *incremental, unsegmented.*
- Subword-level segmentation result of the streaming transcription is unstable.
  - Existing method: remove the last to prevent it from being incomplete.

<table>
<thead>
<tr>
<th>Streaming Transcription</th>
<th>Tokenization of Streaming Transcription Input</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard Wait-2</td>
</tr>
<tr>
<td>他是研究生物的</td>
<td>他/是/</td>
</tr>
<tr>
<td>他是研究生物的</td>
<td>他/是/研/</td>
</tr>
<tr>
<td>他是研究生物的</td>
<td>他/是/研究/</td>
</tr>
<tr>
<td>他是研究生物的</td>
<td>他/是/研究生/</td>
</tr>
<tr>
<td>他是研究生物的</td>
<td>他/是/研究/生物/</td>
</tr>
<tr>
<td>他是研究生物的</td>
<td>他/是/研究/生物/的/</td>
</tr>
</tbody>
</table>
For domain adaptability
- General domain the spoken language domain are quite different:
  - Word order
  - Punctuation
  - Modal particles
  - ...

Our system
- Robust:
  - Propose the Char-Level Wait-k Policy
- Domain adaptation:
  - Apply data augmentation on spoken language domain.
  - Combine two training methods to enhance the predictive ability.
Contents

Motivation

Method

Experiments

Conclusion
Char-Level Wait-k Policy

- **Source**: character sequence after char-level tokenization.
- **Target**: subword sequence after subword-level segmentation and BPE.
- **Read / Write policy**: waiting for $k$ source characters first, and then reading and writing alternately.

### Table

<table>
<thead>
<tr>
<th>Input Sentence</th>
<th>Output Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source</strong></td>
<td><strong>Target</strong></td>
</tr>
<tr>
<td>subword-level MT</td>
<td>subword-level MT</td>
</tr>
<tr>
<td>character-level MT</td>
<td>character-level MT</td>
</tr>
<tr>
<td>char-level tokenization</td>
<td>char-level tokenization</td>
</tr>
</tbody>
</table>

- **S.**: Character-level MT
- **T.**: Subword-level MT

**Method**

Standard wait-k policy:

Char-level wait-k policy:
Why char-level simultaneous translation?

- More robust
  - avoid unstable prefixes caused by subword segmentation.

- More fine-grained latency
  - if one character is enough to express the meaning of a entire word, the ST system does not have to wait for the complete word.

- Translation quality will not be affected too much
  - only performs char-level tokenization on the source, and the target retains subword-level tokenization.
## Domain Adaptation

- **Depunctuation**
  - **Source**: delete the ending punctuation.
  - **Target**: unchanged.

- **Data Augmentation**
  - For spoken language domain corpus.
  - **Source**: we perform 5 data augmentation operations.
  - **Target**: unchanged.

### Data Augmentation Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Original</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Comma</td>
<td>2017年我到北京上大学</td>
<td>2017年我到北京上大学</td>
</tr>
<tr>
<td>Add Tone character</td>
<td>1957年我到北京上大学</td>
<td>1957年我到北京上大学</td>
</tr>
<tr>
<td>Copy Character</td>
<td>1957年我到北京上大学</td>
<td>1957年我到北京上大学</td>
</tr>
<tr>
<td>Replace Homophone</td>
<td>1957年我到北京上大学</td>
<td>1957年我到北京上大学</td>
</tr>
<tr>
<td>Delete Character</td>
<td>1957年我到北京上大学</td>
<td>1957年我到北京上大学</td>
</tr>
</tbody>
</table>
Training Methods

- **Pre-training**: general domain MT corpus
  - Multi-path training (Elbayad et al., 2020)
  - Future-guided training (Zhang et al., 2020b)

- **Fine-tuning**: spoken language domain corpus
  - Original training: fix $k$ and use the original prefix-to-prefix framework for training, and train different models for different $k$.

---

Datasets

- **CWMT19 Chinese → English**: for pre-training.
- **Transcription**: for fine-tuning.
- **Dev. Set**: for evaluation.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Domain</th>
<th>#Sentence Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWMT19</td>
<td>General</td>
<td>9,023,708</td>
</tr>
<tr>
<td>Transcription</td>
<td>Spoken</td>
<td>37,901</td>
</tr>
<tr>
<td>Dev. Set</td>
<td>Spoken</td>
<td>956</td>
</tr>
</tbody>
</table>

System setting

- **Offline**: full-sentence MT based on Transformer.
- **Standard Wait-k**: standard subword-level waitk policy.
- **Standard Wait-k + rm Last Token**: In the inference time, the last token after the word segmentation is remove to prevent it from being incomplete.
- **Char-Level Wait-k**: our proposed method.
Experiments

Main Result

- Char-Level Wait-k improves about 6 BLEU at low latency (AL=1.10).
- More stable and robust.
Experiments

- **Ablation Study**
  - **Data processing**: ‘Depunctuation’ and ‘Data Augmentation’
  - **Training methods**: ‘Future-guided’ and ‘Multi-path’
Conclusion

- The proposed char-level wait-k policy is more robust.
- Data processing and two training methods improve the spoken language domain adaptability.
- For some language pairs with a large length ratio between the source (char) and the target (bpe), we can read multiple characters at each step to deal with the long char-level source. We put this into our future work.
Thanks!

Contact me with:
✉️ zhangshaolei20z@ict.ac.cn