ESPnet -ST: All-in-One Speech Translation Toolkit

Hirofumi Inaguma¹ Shun Kiyono² Kevin Duh³
Shigeki Karita⁴ Nelson Yalta⁵
Tomoki Hayashi⁶,⁷ Shinji Watanabe³

¹Kyoto university  ²RIKEN AIP  ³Johns Hopkins University
⁴NTT Communication Science Laboratories
⁵Waseda University  ⁶Nagoya University
⁷Human Dataware Lab. Co., Ltd.

https://github.com/espnet/espnet
Speech translation: Cascade vs. End-to-end

Cascaded speech translation system
- Modularized training (ASR->MT)
- Still better performance than E2E

End-to-end speech translation (E2E-ST)
- Low-latency during inference
- Mitigate error propagation from ASR
- Easy implementation
- Endangered language documentation

Diagram:
- Source speech
- Speech recognition (ASR)
- Text normalization
- Machine translation (MT)
- Simultaneous interpretation
- Target translation
- Text-to-speech (TTS)
- Target speech
Key features

• ASR/LM/MT/E2E-ST/TTS in a single unified framework
  ➢ PyTorch backend
  ➢ Both cascade and E2E speech translation systems are supported

• Reproducible SOTA results on most corpora

• Support various recipes
  ➢ All you need is to command "./run.sh"
  ➢ Data downloading, data preprocessing, feature extraction, dataset construction, training, decoding

• Provide pre-trained ASR/LM/E2E-ST/MT/TTS models
Comparison with other seq2seq frameworks

- ASR: Wav2letter++, Kaldi
- TTS: ESPnet-ST, OpenSeq2seq, NeMo
- E2E: Fariseq, Open-NMT, Tensor2Tensor, RETURNN, SLT.KIT
- MT: Marian NMT, Sockeye
- FBK-Fairseq-ST, E2E-ST
## Comparison with other seq2seq frameworks

<table>
<thead>
<tr>
<th>Toolkit</th>
<th>Supported task</th>
<th>Example (w/ corpus pre-processing)</th>
<th>Pre-trained model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASR</td>
<td>LM</td>
<td>E2E-ST</td>
</tr>
<tr>
<td>ESPnet-ST (ours)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Lingvo¹</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>OpenSeq2seq²</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>NeMo³</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>RETURNN⁴</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SLT.KIT⁵</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Fairseq⁶</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>Tensor2Tensor⁷</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>OpenNMT-{py, tf}⁸</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>Kaldi⁹</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>Wav2letter++¹⁰</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 1: Framework comparison on supported tasks in January, 2020. ¹Not publicly available. ²Available only in Google Cloud storage. ³(Shen et al., 2019) ⁴(Kuchaiev et al., 2018) ⁵(Kuchaiev et al., 2019) ⁶(Zeyer et al., 2018) ⁷(Zenkel et al., 2018) ⁸(Ott et al., 2019) ⁹(Vaswani et al., 2018) ¹⁰(Klein et al., 2017) ¹¹(Povey et al., 2011) ¹²(Pratap et al., 2019)
Why is the unified framework important?

• E2E-ST model is simple, but difficult to optimize
  ➢ Initialization with well-trained ASR/MT models (pre-training)
  ➢ Auxiliary ASR/MT objective (multi-task learning)

• Combining multiple toolkits for E2E-ST is difficult
  ➢ Parameter namespace does not match across toolkits
  ➢ I/O interfaces are different across toolkits
Pre-training/transfer learning from ASR/MT models

*Multilingual training is also supported

**Speech data augmentation is also supported (speed perturbation, SpecAugment)
Multi-task learning with auxiliary ASR/MT objective

Objective function

\[ \mathcal{L}_{\text{total}} = (1 - \lambda_{\text{asr}} - \lambda_{\text{mt}}) \mathcal{L}_{\text{st}} + \lambda_{\text{asr}} \mathcal{L}_{\text{asr}} + \lambda_{\text{mt}} \mathcal{L}_{\text{mt}} \]
History of ESPnet

Kaldi [Povey et al., 2011]
• Most famous ASR toolkit for conventional HMM-based hybrid system

ESPnet [Watanabe et al., 2018]
• Designed for E2E-ASR systems started from Kaldi-like data preparation
• More than 2.4k stars on GitHub
• Transformer ASR reached SOTA results on Librispeech dataset [Karita et al., 2019]
• Most ASR corpora (w/ pre-trained models + demo) are covered (38 ASR corpora@June 2020)

ESPnet-TTS [Hayashi et al., 2020]
• Extended to end-to-end TTS task
• Most TTS corpora (w/ pre-trained models + demo) are covered (12 TTS corpora@June 2020)
Recipes for multiple corpora

• Existing toolkits do not include corpus preprocessing for ST tasks
  ➢ Manual alignment across ASR/ST/MT tasks are necessary (it’s non-trivial!)

• Most benchmark ST corpora are supported (including corresponding ASR/MT recipes)
  • Fisher-Callhome Spanish (Es->En, 160h)
  • Libri-trans (En->Fr, 100h)
  • Must-C (En->{De, Pt, Fr, Es, Ro, Ru, Nl, It}, 400h)
  • How2 (En->Pt 300h)
  • ST-TED (En->De, 200h)
  • Mboshi-French (Mboshi->Fr, 4h)

• MT corpora
  ➢ All ST corpora
  ➢ IWSLT16 (En<->De)
Installation

• All relevant tools are automatically installed with Makefile
  Docker is also supported

• Neural network library
  - PyTorch
  - warpctc-pytorch

• Preprocessing toolkit
  - Kaldi (speech feature extraction)
  - Moses (text pre-processing, BLEU calculation)
  - Sentencepiece (vocabulary construction) etc.
Stage-by-stage processing

Overall directory structure
Other features

Experiment manager
• Tensorboard
• Chainer reporter
• Attention plotting at every epoch
• Loss/Accuracy/BLEU/PPL plotting at every epoch

Large-scale training/decoding
• Job scheduler with SLURM/Grid Engine etc.
• Multi-GPU training
• Mixed precision training with apex
• Batch beam search decoding (on multiple CPUs)
## Supported model

<table>
<thead>
<tr>
<th>Task</th>
<th>LSTM/GRU</th>
<th>Transformer</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR</td>
<td>○</td>
<td>○</td>
<td>CTC, Joint CTC/Attention, RNN transducer, Transformer transducer, Lightweight and dynamic conv.</td>
</tr>
<tr>
<td>LM</td>
<td>○</td>
<td>○</td>
<td>N-gram</td>
</tr>
<tr>
<td>E2E-ST</td>
<td>○</td>
<td>○</td>
<td>-</td>
</tr>
<tr>
<td>MT</td>
<td>○</td>
<td>○</td>
<td>-</td>
</tr>
<tr>
<td>TTS</td>
<td>○ (Tacotron2)</td>
<td>○</td>
<td>Multi-speaker TTS, voice conversion, FastSpeech (non-autoregressive TTS)</td>
</tr>
</tbody>
</table>
ST results: Must-C (En->8 languages)

- SOTA results only with well-known techniques
- Cascade is still better

Configuration
- Transformer
- ASR pre-training
- MT pre-training
- SpecAugment
- No external data
Text-based MT results: IWSLT16 (En<->De)

Comparable to Fairseq

Configuration
- Transformer
Future work

• Support more MT corpora (e.g., WMT)

• Semi-supervised training
  ➢ Self-supervised learning with pseudo labelling
  ➢ Back translation

• Simultaneous translation

Get started: https://github.com/espnet/espnet
E2E-ST demo