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

Hirofumi Inaguma[1] **Shun Kiyono**[2] **Kevin Duh**[3] **Shigeki Karita**[4]
Nelson Yalta[5] **Tomoki Hayashi**[6,7] **Shinji Watanabe**[3]
[1] Kyoto University [2] RIKEN AIP [3] Johns Hopkins University
[4] NTT Communication Science Laboratories [5] Waseda University
[6] Nagoya University [7] Human Dataware Lab. Co., Ltd.
inaguma@sap.ist.i.kyoto-u.ac.jp

Abstract

We present *ESPnet-ST*, which is designed for the quick development of speech-to-speech translation systems in a single framework. *ESPnet-ST* is a new project inside end-to-end speech processing toolkit, ESPnet, which integrates or newly implements automatic speech recognition, machine translation, and text-to-speech functions for speech translation. We provide all-in-one recipes including data pre-processing, feature extraction, training, and decoding pipelines for a wide range of benchmark datasets. Our reproducible results can match or even outperform the current state-of-the-art performances; these pre-trained models are downloadable. The toolkit is publicly available at https://github.com/espnet/espnet.

1 Introduction

Speech translation (ST), where converting speech signals in a language to text in another language, is a key technique to break the language barrier for human communication. Traditional ST systems involve cascading automatic speech recognition (ASR), text normalization (e.g., punctuation insertion, case restoration), and machine translation (MT) modules; we call this Cascade-ST (Ney, 1999; Casacuberta et al., 2008; Kumar et al., 2014). Recently, sequence-to-sequence (S2S) models have become the method of choice in implementing both the ASR and MT modules (c.f. (Chan et al., 2016; Bahdanau et al., 2015)). This convergence of models has opened up the possibility of designing end-to-end speech translation (E2E-ST) systems, where a single S2S directly maps speech in a source language to its translation in the target language (Bérard et al., 2016; Weiss et al., 2017).

E2E-ST has several advantages over the cascaded approach: (1) a single E2E-ST model can reduce latency at inference time, which is useful for time-critical use cases like simultaneous interpretation. (2) A single model enables back-propagation training in an end-to-end fashion, which mitigates the risk of error propagation by cascaded modules. (3) In certain use cases such as endangered language documentation (Bird et al., 2014), source speech and target text translation (without the intermediate source text transcript) might be easier to obtain, necessitating the adoption of E2E-ST models (Anastasopoulos and Chiang, 2018). Nevertheless, the verdict is still out on the comparison of translation quality between E2E-ST and Cascade-ST. Some empirical results favor E2E (Weiss et al., 2017) while others favor Cascade (Niehues et al., 2019); the conclusion also depends on the nuances of the training data condition (Sperber et al., 2019).

We believe the time is ripe to develop a unified toolkit that facilitates research in both E2E and cascaded approaches. We present *ESPnet-ST*, a toolkit that implements many of the recent models for E2E-ST, as well as the ASR and MT modules for Cascade-ST. Our goal is to provide a toolkit where researchers can easily incorporate and test new ideas under different approaches. Recent research suggests that pre-training, multi-task learning, and transfer learning are important techniques for achieving improved results for E2E-ST (Bérard et al., 2018; Anastasopoulos and Chiang, 2018; Bansal et al., 2019; Inaguma et al., 2019). Thus, a unified toolkit that enables researchers to seamlessly mix-and-match different ASR and MT models in training both E2E-ST and Cascade-ST systems would facilitate research in the field.[1]

ESPnet-ST is especially designed to target the ST task. ESPnet was originally developed for the

[1]There exist many excellent toolkits that support both ASR and MT tasks (see Table 1). However, it is not always straightforward to use them for E2E-ST and Cascade-ST, due to incompatible training/inference pipelines in different modules or lack of detailed preprocessing/training scripts.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 302–311
July 5 - July 10, 2020.

Toolkit	Supported task						Example (w/ corpus pre-processing)						Pre-trained model
	ASR	LM	E2E-ST	Cascade-ST	MT	TTS	ASR	LM	E2E-ST	Cascade-ST	MT	TTS	
ESPnet-ST (ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lingvo[1]	✓	✓	✓♣	✓♣	✓	✓♣	✓	✓	–	–	✓	–	–
OpenSeq2seq[2]	✓	✓	–	–	✓	✓	✓	✓	–	–	✓	–	✓
NeMo[3]	✓	✓	–	–	✓	✓	✓	✓	–	–	✓	–	✓
RETURNN[4]	✓	✓	✓	–	✓	–	–	–	–	–	–	–	✓
SLT.KIT[5]	✓	–	✓	✓	✓	–	✓	–	✓	✓	✓	–	✓
Fairseq[6]	✓	✓	–	–	✓	–	✓	✓	–	–	✓	–	✓
Tensor2Tensor[7]	✓	✓	–	–	✓	–	–	–	–	–	✓	–	✓◇
OpenNMT-{py, tf}[8]	✓	✓	–	–	✓	–	–	–	–	–	–	–	✓
Kaldi[9]	✓	✓	–	–	–	–	✓	✓	–	–	–	–	✓
Wav2letter++[10]	✓	✓	–	–	–	–	✓	✓	–	–	–	–	✓

Table 1: Framework comparison on supported tasks in January, 2020. ♣Not publicly available. ◇Available only in Google Cloud storage. [1](Shen et al., 2019) [2](Kuchaiev et al., 2018) [3](Kuchaiev et al., 2019) [4](Zeyer et al., 2018) [5](Zenkel et al., 2018) [6](Ott et al., 2019) [7](Vaswani et al., 2018) [8](Klein et al., 2017) [9](Povey et al., 2011) [10](Pratap et al., 2019)

ASR task (Watanabe et al., 2018), and recently extended to the text-to-speech (TTS) task (Hayashi et al., 2020). Here, we extend ESPnet to ST tasks, providing code for building translation systems and recipes (i.e., scripts that encapsulate the entire training/inference procedure for reproducibility purposes) for a wide range of ST benchmarks. This is a non-trivial extension: with a unified codebase for ASR/MT/ST and a wide range of recipes, we believe ESPnet-ST is an *all-in-one toolkit* that should make it easier for both ASR and MT researchers to get started in ST research.

The contributions of *ESPnet-ST* are as follows:

- To the best of our knowledge, this is the first toolkit to include ASR, MT, TTS, and ST recipes and models in the same codebase. Since our codebase is based on the unified framework with a common stage-by-stage processing (Povey et al., 2011), it is very easy to customize training data and models.
- We provide recipes for ST corpora such as Fisher-CallHome (Post et al., 2013), Libri-trans (Kocabiyikoglu et al., 2018), How2 (Sanabria et al., 2018), and Must-C (Di Gangi et al., 2019a)[2]. Each recipe contains a single script (`run.sh`), which covers all experimental processes, such as corpus preparation, data augmentations, and transfer learning.
- We provide the open-sourced toolkit and the pre-trained models whose hyper-parameters are intensively tuned. Moreover, we provide interactive demo of speech-to-speech translation hosted by Google Colab.[3]

2 Design

2.1 Installation

All required tools are automatically downloaded and built under `tools` (see Figure 1) by a make command. The tools include (1) neural network libraries such as PyTorch (Paszke et al., 2019), (2) ASR-related toolkits such as Kaldi (Povey et al., 2011), and (3) MT-related toolkits such as Moses (Koehn et al., 2007) and sentencepiece (Kudo, 2018). *ESPnet-ST* is implemented with Pytorch backend.

2.2 Recipes for reproducible experiments

We provide various recipes for all tasks in order to quickly and easily reproduce the strong baseline systems with a single script. The directory structure is depicted as in Figure 1. `egs` contains corpus directories, in which the corresponding task directories (e.g., `st1`) are included. To run experiments, we simply execute `run.sh` under the desired task directory. Configuration yaml files for feature extraction, data augmentation, model training, and decoding etc. are included in `conf`. Model directories including checkpoints are saved under `exp`. More details are described in Section 2.4.

[2]We also support ST-TED (Jan et al., 2018) and low-resourced Mboshi-French (Godard et al., 2018) recipes.

[3]https://colab.research.google.com/github/espnet/notebook/blob/master/st_demo.ipynb

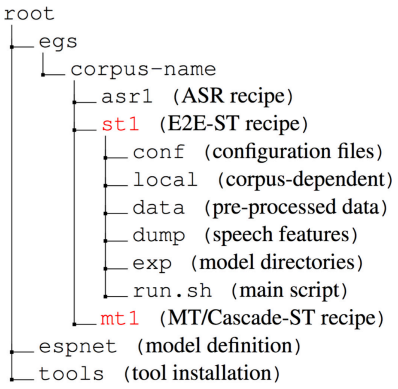

Figure 1: Directory structure of ESPnet-ST

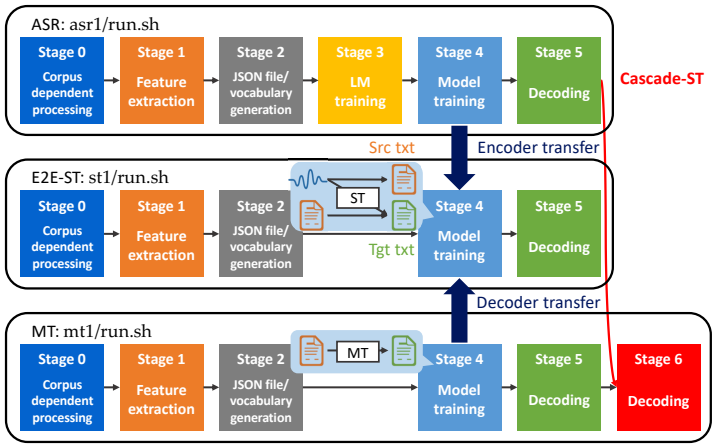

Figure 2: All-in-one process pipelines in ESPnet-ST

2.3 Tasks

We support language modeling (LM), neural text-to-speech (TTS) in addition to ASR, ST, and MT tasks. To the best of our knowledge, none of frameworks support all these tasks in a single toolkit. A comparison with other frameworks are summarized in Table 1. Conceptually, it is possible to combine ASR and MT modules for Cascade-ST, but few frameworks provide such examples. Moreover, though some toolkits indeed support speech-to-text tasks, it is not trivial to switch ASR and E2E-ST tasks since E2E-ST requires the auxiliary tasks (ASR/MT objectives) to achieve reasonable performance.

2.4 Stage-by-stage processing

ESPnet-ST is based on a stage-by-stage processing including corpus-dependent pre-processing, feature extraction, training, and decoding stages. We follow Kaldi-style data preparation, which makes it easy to augment speech data by leveraging other data resources prepared in `egs`.

Once `run.sh` is executed, the following processes are started.

Stage 0: Corpus-dependent pre-processing is conducted using scripts under `local` and the resulting text data is automatically saved under `data`. Both transcriptions and the corresponding translations with three different treatments of casing and punctuation marks (hereafter, punct.) are generated after text normalization and tokenization with `tokenizer.perl` in Moses; (a) **tc**: truecased text with punct., (b) **lc**: lowercased text with punct., and (3) **lc.rm**: lowercased text without punct. except for apostrophe. `lc.rm` is designed for the ASR task since the conventional ASR system does not generate punctuation marks. However, it is possible to train ASR models so as to generate truecased text using `tc`.[4]

Stage 1: Speech feature extraction based on Kaldi and our own implementations is performed.

Stage 2: Dataset JSON files in a format ingestable by ESPnet's Pytorch back-end (containing token/utterance/speaker/language IDs, input and output sequence lengths, transcriptions, and translations) are dumped under `dump`.

Stage 3: (ASR recipe only) LM is trained.

Stage 4: Model training (RNN/Transformer) is performed.

Stage 5: Model averaging, beam search decoding, and score calculation are conducted.

Stage 6: (Cascade-ST recipe only) The system is evaluated by feeding ASR outputs to the MT model.

2.5 Multi-task learning and transfer learning

In ST literature, it is acknowledged that the optimization of E2E-ST is more difficult than individually training ASR and MT models. Multitask training (MTL) and transfer learning from ASR and MT tasks are promising approaches for this problem (Weiss et al., 2017; Bérard et al., 2018; Sperber et al., 2019; Bansal et al., 2019). Thus, in **Stage 4** of the E2E-ST recipe, we allow options to add auxiliary ASR and MT objectives. We also support options to initialize the parameters of the ST encoder with a pre-trained ASR encoder in `asr1`, and to initialize the parameters of the ST decoder with a pre-trained MT decoder in `mt1`.

[4] We found that this degrades the ASR performance.

2.6 Speech data augmentation

We implement techniques that have shown to give improved robustness in the ASR component.

Speed perturbation We augmented speech data by changing the speed with factors of 0.9, 1.0, and 1.1, which results in 3-fold data augmentation. We found this is important to stabilize E2E-ST training.

SpecAugment Time and frequency masking blocks are randomly applied to log mel-filterbank features. This has been originally proposed to improve the ASR performance and shown to be effective for E2E-ST as well (Bahar et al., 2019b).

2.7 Multilingual training

Multilingual training, where datasets from different language pairs are combined to train a single model, is a potential way to improve performance of E2E-ST models (Inaguma et al., 2019; Di Gangi et al., 2019c). Multilingual E2E-ST/MT models are supported in several recipes.

2.8 Additional features

Experiment manager We customize the data loader, trainer, and evaluator by overriding Chainer (Tokui et al., 2019) modules. The common processes are shared among all tasks.

Large-scale training/decoding We support job schedulers (e.g., SLURM, Grid Engine), multiple GPUs and half/mixed-precision training/decoding with `apex` (Micikevicius et al., 2018).[5] Our beam search implementation vectorizes hypotheses for faster decoding (Seki et al., 2019).

Performance monitoring Attention weights and all kinds of training/validation scores and losses for ASR, MT, and ST tasks can be collectively monitored through TensorBoard.

Ensemble decoding Averaging posterior probabilities from multiple models during beam search decoding is supported.

3 Example Models

To give a flavor of the models that are supported with ESPnet-ST, we describe in detail the construction of an example E2E-ST model, which is used later in the Experiments section. Note that there are many customizable options not mentioned here.

[5] https://github.com/NVIDIA/apex

Automatic speech recognition (ASR) We build ASR components with the Transformer-based hybrid CTC/attention framework (Watanabe et al., 2017), which has been shown to be more effective than RNN-based models on various speech corpora (Karita et al., 2019). Decoding with the external LSTM-based LM trained in the **Stage 3** is also conducted (Kannan et al., 2017). The transformer uses 12 self-attention blocks stacked on the two VGG blocks in the speech encoder and 6 self-attention blocks in the transcription decoder; see (Karita et al., 2019) for implementation details.

Machine translation (MT) The MT model consists of the source text encoder and translation decoder, implemented as a transformer with 6 self-attention blocks. For simplicity, we train the MT model by feeding lowercased source sentences without punctuation marks (`lc.rm`) (Peitz et al., 2011). There are options to explore characters and different subword units in the MT component.

End-to-end speech translation (E2E-ST) Our E2E-ST model is composed of the speech encoder and translation decoder. Since the definition of parameter names is exactly same as in the ASR and MT components, it is quite easy to copy parameters from the pre-trained models for transfer learning. After ASR and MT models are trained as described above, their parameters are extracted and used to initialize the E2E-ST model. The model is then trained on ST data, with the option of incorporating multi-task objectives as well.

Text-to-speech (TTS) We also support end-to-end text-to-speech (E2E-TTS), which can be applied after ST outputs a translation. The E2E-TTS model consists of the feature generation network converting an input text to acoustic features (e.g., log-mel filterbank coefficients) and the vocoder network converting the features to a waveform. Tacotron 2 (Shen et al., 2018), Transformer-TTS (Li et al., 2019), FastSpeech (Ren et al., 2019), and their variants such as a multi-speaker model are supported as the feature generation network. WaveNet (van den Oord et al., 2016) and Parallel WaveGAN (Yamamoto et al., 2020) are available as the vocoder network. See Hayashi et al. (2020) for more details.

4 Experiments

In this section, we demonstrate how models from our ESPnet recipes perform on benchmark speech

	Model	Es → En				
		Fisher			CallHome	
		dev	dev2	test	devtest	evltest
E2E	Char RNN + ASR-MTL (Weiss et al., 2017)	48.30	49.10	48.70	16.80	17.40
	ESPnet-ST (Transformer)					
	ASR-MTL (multi-task w/ ASR)	46.64	47.64	46.45	16.80	16.80
	+ MT-MTL (multi-task w/ MT)	47.17	48.20	46.99	17.51	17.64
	ASR encoder init. (①)	46.25	47.11	46.21	17.35	16.94
	+ MT decoder init. (②)	46.25	47.60	46.72	17.62	17.50
	+ SpecAugment (③)	48.94	49.32	48.39	18.83	18.67
	+ Ensemble 3 models (① + ② + ③)	**50.76**	**52.02**	**50.85**	**19.91**	**19.36**
Cascade	Char RNN ASR → Char RNN MT (Weiss et al., 2017)	**45.10**	**46.10**	**45.50**	16.20	16.60
	Char RNN ASR → Char RNN MT (Inaguma et al., 2019)♣	37.3	39.6	38.6	16.8	16.5
	ESPnet-ST					
	Transformer ASR◇ → Transformer MT	41.96	43.46	42.16	19.56	19.82

Table 2: BLEU of ST systems on Fisher-CallHome Spanish corpus. ♣Implemented w/ ESPnet. ◇w/ SpecAugment.

	Model	En → Fr
E2E	Transformer + ASR/MT-trans + KD[1]	17.02
	+ Ensemble 3 models	**17.8**
	Transformer + PT△ + adaptor[2]	16.80
	Transformer + PT△ + SpecAugment[3]	17.0
	RNN + TCEN[4,♣]	17.05
	ESPnet-ST (Transformer)	
	ASR-MTL	15.30
	+ MT-MLT	15.47
	ASR encoder init. (①)	15.53
	+ MT decoder init. (②)	16.22
	+ SpecAugment (③)	16.70
	+ Ensemble 3 models (① + ② + ③)	17.40
Cascade	Transformer ASR → Transformer MT[1]	**17.85**
	ESPnet-ST	
	Transformer ASR◇ → Transformer MT	16.96

Table 3: BLEU of ST systems on Libri-trans corpus. ♣Implemented w/ ESPnet. △Pre-training. ◇w/ SpecAugment. [1](Liu et al., 2019) [2](Bahar et al., 2019a) [3](Bahar et al., 2019b) [4](Wang et al., 2020)

	Model	En → Pt
E2E	RNN (Sanabria et al., 2018)	36.0
	ESPnet-ST	
	Transformer	40.59
	+ ASR-MTL	44.90
	+ MT-MLT	45.10
	Transformer + ASR encoder init. (①)	45.03
	+ MT decoder init. (②)	45.63
	+ SpecAugment (③)	45.68
	+ Ensemble 3 models (① + ② + ③)	**48.04**
Cascade	**ESPnet-ST**	
	Transformer ASR → Transformer MT	44.90

Table 4: BLEU of ST systems on How2 corpus

translation corpora: Fisher-CallHome Spanish En→Es, Libri-trans En→Fr, How2 En→Pt, and Must-C En→8 languages. Moreover, we also performed experiments on IWSLT16 En-De to validate the performance of our MT modules.

All sentences were tokenized with the `tokenizer.perl` script in the Moses toolkit (Koehn et al., 2007). We used the joint source and target vocabularies based on byte pair encoding (BPE) (Sennrich et al., 2016) units. ASR vocabularies were created with English sentences only with `lc.rm`. We report 4-gram BLEU (Papineni et al., 2002) scores with the `multi-bleu.perl` script in Moses. For speech features, we extracted 80-channel log-mel filterbank coefficients with 3-dimensional pitch features using Kaldi, resulting 83-dimensional features per frame. Detailed training and decoding configurations are available in `conf/train.yaml` and `conf/decode.yaml`, respectively.

4.1 Fisher-CallHome Spanish (Es→En)

Fisher-CallHome Spanish corpus contains 170-hours of Spanish conversational telephone speech, the corresponding transcription, as well as the English translations (Post et al., 2013). All punctuation marks except for apostrophe were removed (Post et al., 2013; Kumar et al., 2014; Weiss et al., 2017). We report case-insensitive BLEU on Fisher-{*dev*, *dev2*, *test*} (with four references), and CallHome-{*devtest*, *evltest*} (with a single reference). We used 1k vocabulary for all tasks.

Results are shown in Table 2. It is worth noting that we did not use any additional data resource. Both MTL and transfer learning improved the performance of vanilla Transformer. Our best system with SpecAugment matches the current state-of-the-art performance (Weiss et al., 2017). Moreover, the total training/inference time is much shorter since our E2E-ST models are based on the BPE1k unit rather than characters.[6]

[6]Weiss et al. (2017) trained their model for more than 2.5

	Model	De	Pt	Fr	Es	Ro	Ru	Nl	It
E2E	Transformer + ASR encoder init.[1,♣]	17.30	20.10	26.90	20.80	16.50	10.50	18.80	16.80
	ESPnet-ST (Transformer)								
	ASR encoder/MT decoder init.	22.33	27.26	31.54	27.84	20.91	15.32	26.86	22.81
	+ SpecAugment	**22.91**	**28.01**	**32.69**	**27.96**	**21.90**	**15.75**	**27.43**	**23.75**
Cascade	Transformer → Transformer ASR[1]	18.5	21.5	27.9	22.5	16.8	11.1	22.2	18.9
	ESPnet-ST								
	Transformer ASR → Transformer MT	**23.65**	**29.04**	**33.84**	**28.68**	**22.68**	**16.39**	**27.91**	**24.04**

Table 5: BLEU of ST systems on Must-C corpus. ♣Implemented w/ Fairseq. [1](Di Gangi et al., 2019b)

	En→De			De→En		
Framework	test2012	test2013	test2014	test2012	test2013	test2014
Fairseq	27.73	29.45	25.14	32.25	34.23	29.49
ESPnet-ST	26.92	28.88	24.70	32.19	33.46	29.22

Table 6: BLEU of MT systems on IWSLT 2016 corpus

4.2 Libri-trans (En→ Fr)

Libri-trans corpus contains 236-hours of English read speech, the corresponding transcription, and the French translations (Kocabiyikoglu et al., 2018). We used the clean 100-hours of speech data and augmented translation references with Google Translate for the training set (Bérard et al., 2018; Liu et al., 2019; Bahar et al., 2019a,b). We report case-insensitive BLEU on the *test* set. We used 1k vocabulary for all tasks.

Results are shown in Table 3. Note that all models used the same data resource and are competitive to previous work.

4.3 How2 (En→ Pt)

How2 corpus contains English speech extracted from YouTube videos, the corresponding transcription, as well as the Portuguese translation (Sanabria et al., 2018). We used the official 300-hour subset for training. Since speech features in the How2 corpus is pre-processed as 40-channel log-mel filterbank coefficients with 3-dimensional pitch features with Kaldi in advance, we used them without speed perturbation. We used 5k and 8k vocabularies for ASR and E2E-ST/MT models, respectively. We report case-sensitive BLEU on the *dev5* set.

Results are shown in Table 4. Our systems significantly outperform the previous RNN-based model (Sanabria et al., 2018). We believe that our systems can be regarded as the reliable baselines for future research.

weeks with 16 GPUs, while *ESPnet-ST* requires just 1-2 days with a single GPU. The fast inference of ESPnet-ST can be confirmed in our interactive demo page (RTF 0.7755).

4.4 Must-C (En→ 8 langs)

Must-C corpus contains English speech extracted from TED talks, the corresponding transcription, and the target translations in 8 language directions (De, Pt, Fr, Es, Ro, Ru, Nl, and It) (Di Gangi et al., 2019a). We conducted experiments in all 8 directions. We used 5k and 8k vocabularies for ASR and E2E-ST/MT models, respectively. We report case-sensitive BLEU on the *tst-COMMON* set.

Results are shown in Table 5. Our systems outperformed the previous work (Di Gangi et al., 2019b) implemented with the custermized Fairseq[7] with a large margin.

4.5 MT experiment: IWSLT16 En ↔ De

IWSLT evaluation campaign dataset (Cettolo et al., 2012) is the origin of the dataset for our MT experiments. We used En-De language pair. Specifically, IWSLT 2016 training set for training data, test2012 as the development data, and test2013 and test2014 sets as our test data respectively.

We compare the performance of Transformer model in *ESPnet-ST* with that of Fairseq in Table 6. *ESPnet-ST* achieves the performance almost comparable to the Fairseq. We assume that the performance gap is due to the minor difference in the implementation of two frameworks. Also, we carefully tuned the hyper-parameters for the MT task in the small ST corpora, which is confirmed from the reasonable performances of our Cascaded-ST systems. It is acknowledged that Transformer model is extremely sensitive to the hyper-parameters such as the learning rate and the number of warmup

[7] https://github.com/mattiadg/FBK-Fairseq-ST

steps (Popel and Bojar, 2018). Thus, it is possible that the suitable sets of hyper-parameters are different across frameworks.

5 Conclusion

We presented *ESPnet-ST* for the fast development of end-to-end and cascaded ST systems. We provide various all-in-one example scripts containing corpus-dependent pre-processing, feature extraction, training, and inference. In the future, we will support more corpora and implement novel techniques to bridge the gap between end-to-end and cascaded approaches.

Acknowledgment

We thank Jun Suzuki for providing helpful feedback for the paper.

References

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 82–91.

Parnia Bahar, Tobias Bieschke, and Hermann Ney. 2019a. A comparative study on end-to-end speech to text translation. In *Proceedings of 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2019)*, pages 792–799.

Parnia Bahar, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019b. On using SpecAugment for end-to-end speech translation. In *Proceedings of 16th International Workshop on Spoken Language Translation 2019 (IWSLT 2019)*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 58–68.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *Proceedings of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, pages 6224–6228.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. In *Proceedings of NIPS 2016 End-to-end Learning for Speech and Audio Processing Workshop*.

Steven Bird, Lauren Gawne, Katie Gelbart, and Isaac McAlister. 2014. Collecting bilingual audio in remote indigenous communities. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1015–1024.

F. Casacuberta, M. Federico, H. Ney, and E. Vidal. 2008. Recent efforts in spoken language translation. *IEEE Signal Processing Magazine*, 25(3):80–88.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. In *Conference of european association for machine translation*, pages 261–268.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proceedings of 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, pages 4960–4964.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019a. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 2012–2017.

Mattia A Di Gangi, Matteo Negri, and Marco Turchi. 2019b. Adapting transformer to end-to-end spoken language translation. In *Proceedings of 20th Annual Conference of the International Speech Communication Association (INTERSPEECH 2019)*, pages 1133–1137.

Mattia Antonino Di Gangi, Matteo Negri, and Marco Turchi. 2019c. One-to-many multilingual end-to-end speech translation. In *Proceedings of 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2019)*, pages 585–592.

Pierre Godard, Gilles Adda, Martine Adda-Decker, Juan Benjumea, Laurent Besacier, Jamison Cooper-Leavitt, Guy-Noel Kouarata, Lori Lamel, Hélène Maynard, Markus Mueller, Annie Rialland, Sebastian Stueker, François Yvon, and Marcely Zanon-Boito. 2018. A very low resource language speech corpus for computational language documentation experiments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tomoki Hayashi, Ryuichi Yamamoto, Katsuki Inoue, Takenori Yoshimura, Shinji Watanabe, Tomoki Toda,

Kazuya Takeda, Yu Zhang, and Xu Tan. 2020. ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*.

Hirofumi Inaguma, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2019. Multilingual end-to-end speech translation. In *Proceedings of 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2019)*, pages 570–577.

Niehues Jan, Roldano Cattoni, Stüker Sebastian, Mauro Cettolo, Marco Turchi, and Marcello Federico. 2018. The IWSLT 2018 evaluation campaign. In *Proceedings of 15th International Workshop on Spoken Language Translation 2018 (IWSLT 2018)*, pages 2–6.

Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhifeng Chen, and Rohit Prabhavalkar. 2017. An analysis of incorporating an external language model into a sequence-to-sequence model. In *Proceedings of 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, pages 5824–5828.

Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. 2019. A comparative study on Transformer vs RNN in speech applications. In *Proceedings of 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2019)*, pages 499–456.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Ali Can Kocabiyikoglu, Laurent Besacier, and Olivier Kraif. 2018. Augmenting Librispeech with French translations: A multimodal corpus for direct speech translation evaluation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Oleksii Kuchaiev, Boris Ginsburg, Igor Gitman, Vitaly Lavrukhin, Carl Case, and Paulius Micikevicius. 2018. OpenSeq2Seq: Extensible toolkit for distributed and mixed precision training of sequence-to-sequence models. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 41–46.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, et al. 2019. NeMo: a toolkit for building AI applications using Neural Modules. *arXiv preprint arXiv:1909.09577*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 66–75.

Gaurav Kumar, Matt Post, Daniel Povey, and Sanjeev Khudanpur. 2014. Some insights from translating conversational telephone speech. In *Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 3231–3235.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6706–6713.

Yuchen Liu, Hao Xiong, Zhongjun He, Jiajun Zhang, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-end speech translation with knowledge distillation. In *Proceedings of 20th Annual Conference of the International Speech Communication Association (INTERSPEECH 2019)*, pages 1128–1132.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*.

Hermann Ney. 1999. Speech translation: Coupling of recognition and translation. In *Proceedings of 1999 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 1999)*, pages 517–520.

J. Niehues, R. Cattoni, S. Stüker, M. Negri, M. Turchi, E. Salesky, R. Sanabria, L. Barrault, L. Specia, and M Federico. 2019. The IWSLT 2019 evaluation campaign. In *Proceedings of 16th International Workshop on Spoken Language Translation 2019 (IWSLT 2019)*.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible

toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 311–318.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 8024–8035.

Stephan Peitz, Markus Freitag, Arne Mauser, and Hermann Ney. 2011. Modeling punctuation prediction as machine translation. In *Proceedings of 8th International Workshop on Spoken Language Translation 2011 (IWSLT 2011)*, pages 238–245.

Martin Popel and Ondřej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus. In *Proceedings of 10th International Workshop on Spoken Language Translation 2013 (IWSLT 2013)*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *Proceedings of 2011 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2011)*.

Vineel Pratap, Awni Hannun, Qiantong Xu, Jeff Cai, Jacob Kahn, Gabriel Synnaeve, Vitaliy Liptchinsky, and Ronan Collobert. 2019. Wav2Letter++: A fast open-source speech recognition system. In *Proceedings of 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019)*, pages 6460–6464.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 3165–3174.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: A large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*.

Hiroshi Seki, Takaaki Hori, Shinji Watanabe, Niko Moritz, and Jonathan Le Roux. 2019. Vectorized Beam Search for CTC-Attention-Based Speech Recognition. In *Proceedings of 20th Annual Conference of the International Speech Communication Association (INTERSPEECH 2019)*, pages 3825–3829.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725.

Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, Mia X Chen, Ye Jia, Anjuli Kannan, Tara Sainath, Yuan Cao, Chung-Cheng Chiu, et al. 2019. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *arXiv preprint arXiv:1902.08295*.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2018. Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions. In *Proceedings of 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, pages 4779–4783.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. 2019. Attention-passing models for robust and data-efficient end-to-end speech translation. *Transactions of the Association for Computational Linguistics*, 7:313–325.

Seiya Tokui, Ryosuke Okuta, Takuya Akiba, Yusuke Niitani, Toru Ogawa, Shunta Saito, Shuji Suzuki, Kota Uenishi, Brian Vogel, and Hiroyuki Yamazaki Vincent. 2019. Chainer: A deep learning framework for accelerating the research cycle. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2019)*, pages 2002–2011.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 193–199, Boston, MA. Association for Machine Translation in the Americas.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2020. Bridging the gap between pretraining and fine-tuning for end-to-end speech translation. In *Proceedings of the AAAI conference on artificial intelligence 2020 (AAAI 2020)*.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson En-

rique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. 2018. ESPnet: End-to-end speech processing toolkit. In *Proceedings of 19th Annual Conference of the International Speech Communication Association (INTERSPEECH 2018)*, pages 2207–2211.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *Proceedings of 18th Annual Conference of the International Speech Communication Association (INTERSPEECH 2017)*, pages 2625–2629.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*.

Thomas Zenkel, Matthias Sperber, Jan Niehues, Markus Müller, Ngoc-Quan Pham, Sebastian Stüker, and Alex Waibel. 2018. Open source toolkit for speech to text translation. *Prague Bull. Math. Linguistics*, 111:125–135.

Albert Zeyer, Tamer Alkhouli, and Hermann Ney. 2018. RETURNN as a generic flexible neural toolkit with application to translation and speech recognition. In *Proceedings of ACL 2018, System Demonstrations*, pages 128–133.