NeurST: Neural Speech Translation Toolkit

Chengqi Zhao Mingxuan Wang Qianqian Dong Rong Ye Lei Li
ByteDance AI Lab, Shanghai, China
{zhaochengqi.d,wangmingxuan.89,dongqianqian,yerong,lileilab}@bytedance.com

Abstract

NeurST is an open-source toolkit for neural speech translation. The toolkit mainly focuses on end-to-end speech translation, which is easy to use, modify, and extend to advanced speech translation research and products. NeurST aims at facilitating the speech translation research for NLP researchers and building reliable benchmarks for this field. It provides step-by-step recipes for feature extraction, data preprocessing, distributed training, and evaluation. In this paper, we will introduce the framework design of NeurST and show experimental results for different benchmark datasets, which can be regarded as reliable baselines for future research. The toolkit is publicly available at https://github.com/bytedance/neurst and we will continuously update the performance of NeurST with other counterparts and studies at https://st-benchmark.github.io/.

1 Introduction

Speech translation (ST), which translates audio signals of speech in one language into text in a foreign language, is a hot research subject nowadays and has widespread applications, like cross-language videoconferencing or customer support chats.

Traditionally, researchers build a speech translation system via a cascading manner, including an automatic speech recognition (ASR) and a machine translation (MT) subsystem (Ney, 1999; Casacuberta et al., 2008; Kumar et al., 2014). Cascade systems, however, suffer from error propagation problems, where an inaccurate ASR output would theoretically cause translation errors. Owing to recent progress of sequence-to-sequence modeling for both neural machine translation (NMT) (Bahdanau et al., 2015; Luong et al., 2015; Vaswani et al., 2017) and end-to-end speech recognition (Chan et al., 2016; Chiu et al., 2018; Dong et al., 2018), it becomes feasible and efficient to train an end-to-end direct ST model (Berard et al., 2016; Duong et al., 2016; Weiss et al., 2017). This end-to-end fashion attracts much attention due to its appealing properties: *a*) modeling without intermediate ASR transcriptions obviously alleviates the propagation of errors; *b*) a single and unified ST model is beneficial to deployment with lower latency in contrast to cascade systems.

Recent studies show that end-to-end ST models achieve promising performance and are comparable with cascaded models (Ansari et al., 2020). The end-to-end solution has great potential to be the dominant technology for speech translation, however challenges remain. The first is about benchmarks. Many ST studies conduct experiments on different datasets. Liu et al. (2019) evaluate the method on TED English-Chinese; and Dong et al. (2021) use *libri-trans* English-French and IWSLT2018 English-German dataset; and Wu et al. (2020) show the results on CoVoST dataset and the FR/RO portions of MuST-C dataset. Different datasets make it difficult to compare the performance of their approaches. Further, even for the same dataset, the baseline results are not necessarily kept consistent. Take the *libri-trans* English-French dataset as an example. Dong et al. (2021) report the pre-trained baseline as 15.3 and the result of Liu et al. (2019) is 14.3 in terms of tokenized BLEU, while Inaguma et al. (2020) report 15.5 (detokenized BLEU). The mismatching baseline results in an unfair comparison on the improvements of their approaches. We think one of the primary reasons is that the preprocessing of audio data is complex, and the ST model training involves many tricks, such as pre-training and data augmentation.

Therefore a reproducible and reliable benchmark is required. In this work, we present NeurST , a toolkit for easily building and training end-to-end ST models, as well as end-to-end ASR and

NMT for cascade systems. We implement state-of-the-art Transformer-based models (Vaswani et al., 2017; Karita et al., 2019) and provide step-by-step recipes for feature extraction, data preprocessing, model training, and inference for researchers to reproduce the benchmarks. Though there exist several counterparts, such as *Lingvo* (Shen et al., 2019), *fairseq-ST* (Wang et al., 2020a) and *Kaldi* [1] style *ESPnet-ST* (Inaguma et al., 2020), NeurST is specially designed for speech translation tasks, which encapsulates the details of speech processing and frees the developers from data engineering. It is easy to use and extend. The contributions of this work are as follows:

- NeurST is designed specifically for end-to-end ST, with clean and simple code. It is lightweight and independent of *Kaldi*, which simplifies installation and usage, and is more compatible for NLP researchers.
- We report strong benchmarks with well-designed hyper-parameters and show best practice on several ST corpora. We provide a series of recipes to reproduce them, which serves as reliable baselines for the speech translation field.

2 Design and Features

NeurST is implemented with both TensorFlow2 and PyTorch backends. In this section, we will introduce the design components and features of this toolkit.

2.1 Design

NeurST divides one running job into four components: `Dataset`, `Model`, `Task` and `Executor`.

Dataset NeurST abstracts out a common interface `Dataset` for data input. For example, we can train a speech translation model from either a raw dataset tarball or pre-extracted record files. The `Dataset` iterates on the data files and standardizes the read records, e.g., ST tasks only accept key-value pairs storing audio signals/features and translations. One can implement their logic to accept the data of various modalities.

Model NeurST provides an optimal implementation of Transformer and its adaptation to speech-to-text tasks, which achieve state-of-the-art performance on standard benchmarks. Moreover, one can customize various models using TensorFlow2/PyTorch APIs or combine the encoders, decoders, and layers inside the NeurST .

Task NeurST abstracts out `Task` interface to bridge `Dataset` and `Model`. In detail, `Task` defines data pipelines to match the data samples from `Dataset` to the input formats of `Model`. For examples, ST task does tokenization on the text translations and transforms each token to index. In this way, user-defined `Dataset` and `Model` can be efficiently integrated into NeurST , as long as they share the same `Task`.

Executor NeurST provides the execution logic for handling basic workflows of training, validation, and inference. Researchers can either define their specific process of training and evaluation, or pay less attention to API details in `Executor` but reuse them by simply customizing `Dataset`, `Model` and `Task`.

2.2 Features

Computation NeurST has high computation efficiency and it can be further optimized by enabling mixed-precision (Micikevicius et al., 2018) and XLA (Accelerated Linear Algebra). Furthermore, NeurST supports fast distributed training using *Horovod* (Sergeev and Balso, 2018) and *Byteps* (Peng et al., 2019; Jiang et al., 2020) on large-scale scenarios.

Data Preprocessing NeurST supports on-the-fly data preprocessing via a number of lightweight python packages, like python_speech_features[2] for extracting audio features (e.g. mel-frequency cepstral coefficients and log-mel filterbank coefficients). And for text processing, NeurST integrates some effective tokenizers, including moses tokenizer[3], byte pair encoding (BPE) (Sennrich et al., 2016b) and SentencePiece[4]. Alternatively, the training data can be preprocessed and stored in binary files (e.g., TFRecord) beforehand, which is guaranteed to improve the I/O performance during training. Moreover, to simplify such operations, NeurST provides the command-line tool to create such record files, which automatically iterates on various data formats defined by `Dataset`, preprocesses data samples according to `Task` and writes to the disk.

[1] https://kaldi-asr.org/

[2] https://github.com/jameslyons/python_speech_features

[3] The python version: https://github.com/alvations/sacremoses

[4] https://github.com/google/sentencepiece

Transfer Learning NeurST supports initializing the model variables from well-trained models as long as they have the same variable names. As for ST, we can initialize the ST encoder with a well-trained ASR encoder and initialize the ST decoder with a well-trained MT decoder, which facilitates to achieve promising improvements. Besides, NeurST also provides scripts for converting released models from other repositories, like wav2vec2.0 (Baevski et al., 2020) and BERT (Devlin et al., 2019). Researchers can conveniently integrate these pre-trained components to the customized models.

Simultaneous Translation NeurST keeps up with the recent progress of simultaneous translation. The models are extended to train with streaming audio or text input.

Validation while Training NeurST supports customizing validation process during training. By default, NeurST offers evaluation on development data during training and keeps track of the checkpoints with the best evaluation results.

Monitoring NeurST supports TensorBoard for monitoring metrics during training, such as training loss, training speed, and evaluation results.

Model Serving There is no gap between the research models and production models under NeurST , while they can be easily served with TensorFlow Serving. Moreover, for higher performance serving of standard transformer models, NeurST is able to integrate with other optimized inference libraries, like *lightseq* (Wang et al., 2021).

3 Speech Translation Benchmarks

We conducted experiments on several benchmark speech translation corpora using NeurST and compared the performance with other open-source codebases and studies. Though that would be an unfair comparison due to the different model structures and hyperparameters, the goal of NeurST is to provide strong and reproducible benchmarks for future research.

3.1 Datasets

We choose the following publicly available speech translation corpora that include speech in a source language aligned to text in a target language:

task	init scale	end scale	decay at	decay steps
MT	1.0	1.0	-	-
ASR	3.5	2.0	50k	50k
ST	3.5	1.5	50k	50k

Table 1: Hyperparameters of the learning rate schedule. Take the case of ST, the learning rate is scaled up by 3.5x for the first 50k steps. Then, we linearly decrease the scaling factor to 1.5 for 50k steps.

libri-trans (Kocabiyikoglu et al., 2018) [5] is a small EN→FR dataset which was originally started from the *LibriSpeech* corpus, the audiobook recordings for ASR (Panayotov et al., 2015). The English utterances were automatically aligned to the e-books in French, and 236 hours of English speech aligned to French translations at utterance level were finally extracted. It has been widely used in previous studies. As such, we use the clean 100-hour portion plus the augmented machine translation from Google Translate as the training data and follow its split of dev and test data.

MuST-C (Di Gangi et al., 2019)[6] is a multilingual speech translation corpus from English to 8 languages: Dutch (NL), French (FR), German (DE), Italian (IT), Portuguese (PT), Romanian (RO), Russian (RU) and Spanish (ES). MuST-C comprises at least 385 hours of audio recordings from English TED talks with their manual transcriptions and translations at sentence level for training, and we use the *dev* and *tst-COMMON* as our development and test data, respectively. To the best of our knowledge, MuST-C is currently the largest speech translation corpus available for each language pair.

3.2 Data Preprocessing

Beyond the officially released version, we performed no other audio to text alignment and data cleaning on *libri-trans* and MuST-C datasets.

For speech features, we extracted 80-channel logmel filterbank coefficients with windows of 25ms and steps of 10ms, resulting in 80-dimensional features per frame. The audio features of each sample were then normalized by the mean and the standard deviation. All texts were segmented into subword level by first applying Moses tokenizer and then BPE. In detail, we removed all punctuations and lowercased the sentences in the source side while the cases and punctuations of target sentences were

[5] https://github.com/alicank/Translation-Augmented-LibriSpeech-Corpus

[6] https://ict.fbk.eu/must-c/

	Model	tok	detok
Cascade	*ESPnet-ST* ASR *transf-s* + CTC → MT (Inaguma et al., 2020)†	-	17.0
	NeurST ASR *transf-s* → MT	18.2	16.8
End-to-End	ST BiLSTM (Bahar et al., 2019)	17.0	16.2
	ST *transf-s* (Liu et al., 2019)	14.3	-
	ST *transf-s* + KD (Liu et al., 2019)	17.0	-
	ESPnet-ST ST *transf-s* (Inaguma et al., 2020)†	-	16.7
	TCEN-LSTM (Wang et al., 2020b)$^\flat$	-	17.1
	ST *transf-s* (Wang et al., 2020c)	16.0	-
	ST *transf-s* + curriculum pre-training (Wang et al., 2020c)	17.7	-
	LUT (Dong et al., 2021)	17.8	-
	NeurST ST *transf-s*	18.7	17.2

Table 2: Case-insensitive BLEU scores on *libri-trans* test set under constrained setting (without additional ASR and MT data). †Notably, we refer to the results presented in `espnet/egs/libri_trans/st1` and consider them as detokenized BLEU according to the evaluation script in the repository[7]. $^\flat$ The result of TCEN-LSTM is also marked as detokenized BLEU due to its implementation on *ESPnet-ST*.

	Model	DE	ES	FR	IT	NL	PT	RO	RU	avg.
Cascade	ESPnet-ST ASR *transf-s* + CTC → MT (Inaguma et al., 2020)	23.7	28.7	33.8	24.0	27.9	29.0	22.7	16.4	25.8
	NeurST ASR *transf-s* → MT	23.4	28.0	33.9	23.8	27.1	28.3	22.2	16.0	25.3
End-to-End	*ESPnet-ST* ST *transf-s* (Inaguma et al., 2020)	22.9	28.0	32.8	23.8	27.4	28.0	21.9	15.8	25.1
	fairseq-ST ST *transf-s* (Wang et al., 2020a)	22.7	27.2	32.9	22.7	27.3	28.1	21.9	15.3	24.8
	ST *transf-base* + AFS^{t,f} (Zhang et al., 2020)	22.4	26.9	31.6	23.0	24.9	26.3	21.0	14.7	23.9
	NeurST ST *transf-s*	22.8	27.4	33.3	22.9	27.2	28.7	22.2	15.1	24.9

Table 3: Case-sensitive detokenized BLEU scores on MuST-C *tst-COMMON*.

reserved. The BPE rules were jointly learned with 8,000 merge operations and shared across ASR, MT, and ST tasks.

3.3 Benchmark Models

We implemented Transformer (Vaswani et al., 2017), the state-of-the-art sequence-to-sequence model, for all our tasks.

In detail, for MT in cascade systems, the model included 6 layers for both encoder and decoders. The embedding dimension was 256, and the size of hidden units in feedforward layer was 2,048. The attention head for self-attention and cross-attention was set to 4. We used Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.98$ and applied the same schedule algorithm as Vaswani et al. (2017) for learning rate. We trained the MT models with a global batch size of 25,000 tokens.

As for ASR/ST, we referred to the recent progress of Transformer-based end-to-end ASR models (Dong et al., 2018; Karita et al., 2019) and extended the basic transformer model to be compatible with audio inputs. The audio frames were first compressed by two-layer CNN with 256 channels, 3×3 kernel and stride size 2, each of which was followed by a layer normalization. Then, we performed a linear transformation on the compressed audio representations to match the width of the transformer model. We used the same model structure as MT, except that we enlarged the number of encoder layers to 12 to obtain better performance. This configuration is labeled as *transf-s* (transformer small). For training, we used the same Adam optimizer as MT but set the warmup steps to 25,000, and we empirically scaled up the learning rate to accelerate the convergence. The hyperparameters of the learning rate schedule are listed in Table 1. Moreover, for GPU memory efficiency, we truncated the audio frames to 3,000 and removed training samples whose transcription length exceeded 120 and 150 for ASR and ST, respectively. The ASR models were trained with 120,000 frames per batch, while the batch size for ST was 80,000 frames. To further improve the performance of ST, we applied SpecAugment technique (Park et al., 2019) with frequency masking ($mF = 2, F = 27$)

[7] `multi-bleu-detok.perl` in https://github.com/espnet/espnet/blob/master/utils/score_bleu.sh

Model	tok	detok
Cascade		
NeurST ASR *transf-s* → MT	17.4	16.0
End-to-End		
NeurST ST *transf-s*	17.8	16.3
ST *transf-base* + AFS$^{t,f\diamondsuit}$	18.6	17.2

Table 4: Case-sensitive BLEU scores on *libri-trans* test set under constrained setting. $^{\diamondsuit}$is from Zhang et al. (2020) with the proposed adaptive feature selection method, which uses the transformer base setting (embedding size=512).

Model	NeurST	*ESPnet-ST*
ST + ASR enc init.	16.5	15.5
+ MT dec init.	16.6	16.2
+ SpecAug.	17.2	16.7
ST + ASR enc init. + SpecAug.	17.2	-

Table 5: Case-insensitive detokenized BLEU scores on *libri-trans* test set with difference setups.

Model	NeurST	*ESPnet-ST*
pure ST	18.6	-
+ ASR enc init.	21.9	21.8
+ MT dec init.	22.1	22.3
+ SpecAug.	23.3	22.9
ST + ASR enc init. + SpecAug.	22.8	-

Table 6: Case-sensitive detokenized BLEU scores on MuST-C EN-DE *tst-COMMON* with difference setups.

and time masking ($mT = 2, T = 70, p = 0.2$).

Additionally, we applied label smoothing of value 0.1 for training all three tasks. The encoder of the ST model is initialized by the ASR encoder by default unless noted.

3.4 Evaluation

For evaluation, we averaged the latest 10 checkpoints and used a beam width of 4 with no length penalty for all the above tasks.

We use word error rate (WER) to evaluate ASR models and report case-sensitive detokenized BLEU[8] for MT and ST models. In order to compare with existing works, we also report case-insensitive tokenized BLEU using `multi-bleu.perl` in Moses for *libri-trans* dataset.

3.5 Main Results

The overall results and comparisons with other studies are illustrated in Table 2 and 3. It is worth noting that all results are from single models rather than ensemble models.

To make a fair comparison on *libri-trans* corpus, we list both tokenized and detokenized BLEU scores in Table 2 and strive to distinguish the metric of existing literature. Our transformer-based ST model, which only applies ASR pre-training and SpecAugment, achieves superior results versus recent works about knowledge distillation (Liu et al., 2019), curriculum pre-training (Wang et al., 2020c), and LUT (Dong et al., 2021). Compared with the counterpart *ESPnet-ST*, we also outperform by 0.5 BLEU, even though Inaguma et al. (2020) apply additional techniques like speed perturbation, pretrained MT decoder, and CTC loss for ASR pretraining. The cascade baseline is slightly worse than that of *ESPnet-ST* (-0.2 BLEU) because the ASR+CTC can achieve lower WER (6.4)[9] while our pure end-to-end ASR obtains 8.8. We surprisingly find that the end-to-end ST model exceeds the cascade system by 0.4∼0.5 BLEU. We will discuss this in detail in section 3.7. And as a supplementary benchmark, we present case-sensitive BLEU scores in Table 4.

Table 3 illustrates the results on MuST-C *tst-COMMON*. The results of our end-to-end ST model are competitive with both *fairseq-ST* and *ESPnet-ST*.

3.6 Ablation Study

Training a direct ST model is more complicated than training an ASR or MT model. Our preliminary experiment based on a pure end-to-end ST model fails to converge on *libri-trans* corpus, which can be the result of the data scarcity. To alleviate this problem, pre-training some parts of the neural network is the most effective way and has been validated in all existing end-to-end ST studies. We show our results in Table 5 and 6 as a reference for future works. It turns out that we can obtain a reasonable or even better BLEU score by simply initializing the ST encoder with a pre-trained ASR encoder. The improvement by MT decoder initialization is relatively marginal in our setup. Furthermore, the SpecAugment technique can consistently boost ST models.

[8] https://github.com/mjpost/sacrebleu

[9] from https://github.com/espnet/espnet/blob/master/egs/libri_trans/asr1/RESULTS.md

Model	BLEU
large MT (w/ punc. & cased)	36.2
large MT (w/o punc.& lc)	34.3
large cascade ST	31.4
large end-to-end ST	29.7

Table 7: Case-sensitive detokenized BLEU scores on MuST-C EN-DE *tst-COMMON*.

3.7 Cascade versus End-to-End

Previous experiments on *libri-trans* and MuST-C NL/PT show that the end-to-end systems have outperformed the cascade systems. Here we argue that the performance of the cascade systems above is hampered by a lack of quantitative data, and they should take advantage of large amounts of ASR and MT data separately. Hence, we further extended NeurST to large-scale scenarios and experimented on the allowed datasets for IWSLT 2021 evaluation campaign[10]. We followed the practice of Zhao et al. (2021) to build our large cascade and end-to-end ST systems, which contains large-scale back-translation (Sennrich et al., 2016a) and pseudo labeling (also known as knowledge distillation) technologies. The results are illustrated in Table 7. As seen, there is a significant loss of 1.7 BLEU between end-to-end ST and cascade ST. And the cascade system would have the potential to narrow the gap to the pure MT system by introducing extra punctuation restoration and true-case modules.

Though the cascade system is superior under large data conditions, we believe future researches on self-supervised learning, knowledge distillation, and dataset construction would realize the potential of end-to-end models.

4 Conclusion

We introduce NeurST toolkit for easily building and training end-to-end speech translation models. We provide straightforward recipes for audio data pre-processing, training, and inference, which we believe is friendly with NLP researchers. Moreover, we report strong and reproducible benchmarks and will continuously catch up on advanced progress using NeurST , which can be regarded as the reliable baselines for the ST field.

[10] https://iwslt.org/2021/offline

References

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online. Association for Computational Linguistics.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Francisco Casacuberta, Marcello Federico, Hermann Ney, and Enrique Vidal. 2008. Recent efforts in spoken language translation. *IEEE Signal Process. Mag.*, 25(3):80–88.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 4960–4964. IEEE.

Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 4774–4778. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. 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, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 5884–5888. IEEE.

Qianqian Dong, Rong Ye, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. 2021. Listen, understand and translate: Triple supervision decouples end-to-end speech-to-text translation. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. An attentional model for speech translation without transcription. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 949–959, San Diego, California. Association for Computational Linguistics.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. 2020. ESPnet-ST: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 302–311, Online. Association for Computational Linguistics.

Yimin Jiang, Yibo Zhu, Chang Lan, Bairen Yi, Yong Cui, and Chuanxiong Guo. 2020. A unified architecture for accelerating distributed DNN training in heterogeneous gpu/cpu clusters. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, pages 463–479.

Shigeki Karita, Xiaofei Wang, Shinji Watanabe, Takenori Yoshimura, Wangyou Zhang, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, and Ryuichi Yamamoto. 2019. A comparative study on transformer vs RNN in speech applications. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019*, pages 449–456.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-End Speech Translation with Knowledge Distillation. In *Proc. Interspeech 2019*, pages 1128–1132.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 5206–5210. IEEE.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 2613–2617.

Yanghua Peng, Yibo Zhu, Yangrui Chen, Yixin Bao, Bairen Yi, Chang Lan, Chuan Wu, and Chuanxiong Guo. 2019. A generic communication scheduler for distributed DNN training acceleration. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles, SOSP 2019*, pages 16–29.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Alexander Sergeev and Mike Del Balso. 2018. Horovod: fast and easy distributed deep learning in tensorflow. *CoRR*, abs/1802.05799.

Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, Mia Xu Chen, Ye Jia, Anjuli Kannan, Tara N. Sainath, Yuan Cao, Chung-Cheng Chiu, Yanzhang He, Jan Chorowski, Smit Hinsu, Stella Laurenzo, James Qin, Orhan Firat, Wolfgang Macherey, Suyog Gupta, Ankur Bapna, Shuyuan Zhang, Ruoming Pang, Ron J. Weiss, Rohit Prabhavalkar, Qiao Liang, Benoit Jacob, Bowen Liang, HyoukJoong Lee, Ciprian Chelba, Sébastien Jean, Bo Li, Melvin Johnson, Rohan Anil, Rajat Tibrewal, Xiaobing Liu, Akiko Eriguchi, Navdeep Jaitly, Naveen Ari, Colin Cherry, Parisa Haghani, Otavio Good, Youlong Cheng, Raziel Alvarez, Isaac Caswell, Wei-Ning Hsu, Zongheng Yang, Kuan-Chieh Wang, Ekaterina Gonina, Katrin Tomanek, Ben Vanik, Zelin Wu, Llion Jones, Mike Schuster, Yanping Huang, Dehao Chen, Kazuki Irie, George F. Foster, John Richardson, and et al. 2019. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *CoRR*, abs/1902.08295.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020a. Fairseq S2T: Fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 33–39, Suzhou, China. Association for Computational Linguistics.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2020b. Bridging the gap between pre-training and fine-tuning for end-to-end speech translation. In *AAAI*, pages 9161–9168. AAAI Press.

Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020c. Curriculum pre-training for end-to-end speech translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3728–3738, Online. Association for Computational Linguistics.

Xiaohui Wang, Ying Xiong, Yang Wei, Mingxuan Wang, and Lei Li. 2021. LightSeq: A high performance inference library for transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 113–120, Online. Association for Computational Linguistics.

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

Anne Wu, Changhan Wang, Juan Pino, and Jiatao Gu. 2020. Self-supervised representations improve end-to-end speech translation. In *Interspeech 2020*.

Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. 2020. Adaptive feature selection for end-to-end speech translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2533–2544, Online. Association for Computational Linguistics.

Chengqi Zhao, Zhicheng Liu, Jian Tong, Tao Wang, Mingxuan Wang, Rong Ye, Qianqian Dong, Jun Cao, and Lei Li. 2021. The volctrans neural speech translation system for IWSLT 2021. *CoRR*, abs/2105.07319.