The 2020 ESPnet Update: New Features, Broadened Applications, Performance Improvements, and Future Plans

Watanabe, Shinji; Boyer, Florian; Chang, Xuankai; Guo, Pengcheng; Hayashi, Tomoki; Higuchi, Yosuke; Hori, Takaaki; Huang, Wen-Chin; Inaguma, Hirofumi; Kamo, Naoyuki; Shigeki, Karita; Li, Chenda; Shi, Jing; Subramanian, Aswin S; Zhang, Wangyou

Abstract

This paper describes the recent development of ESPnet (https://github.com/espnet/espnet), an end-to-end speech processing toolkit. This project was initiated in December 2017 to mainly deal with end-to-end speech recognition experiments based on sequence-to-sequence modeling. The project has grown rapidly and now covers a wide range of speech processing applications. Now ESPnet also includes text to speech (TTS), voice conversation (VC), speech translation (ST), and speech enhancement (SE) with support for beamforming, speech separation, denoising, and dereverberation. All applications are trained in an end-to-end manner, thanks to the generic sequence to sequence modeling properties, and they can be further integrated and jointly optimized. Also, ESPnet provides reproducible all-in-one recipes for these applications with state-of-the-art performance in various benchmarks by incorporating transformer, advanced data augmentation, and conformer. This project aims to provide up-to-date speech processing experience to the community so that researchers in academia and various industry scales can develop their technologies collaboratively.

IEEE Data Science and Learning Workshop (DSLW)

THE 2020 ESPNET UPDATE: NEW FEATURES, BROADENED APPLICATIONS, PERFORMANCE IMPROVEMENTS, AND FUTURE PLANS

Shinji Watanabe[1], Florian Boyer[2,3], Xuankai Chang[1], Pengcheng Guo[4,1], Tomoki Hayashi[5,6]
Yosuke Higuchi[7], Takaaki Hori[8], Wen-Chin Huang[6], Hirofumi Inaguma[9], Naoyuki Kamo[10],
Shigeki Karita[11], Chenda Li[12], Jing Shi[13], Aswin Shanmugam Subramanian[1], Wangyou Zhang[12]

[1]Johns Hopkins University, [2]Airudit, Speech Lab., [3]LaBRI, Bordeaux INP, CNRS, UMR 5800,
[4]Northwestern Polytechnical University, [5]Human Dataware Lab. Co., Ltd., [6]Nagoya University
[7]Waseda University, [8]MERL, [9]Kyoto University, [10]NTT Corporation, [11]Google
[12]Shanghai Jiao Tong University, [13]Institute of Automation, Chinese Academy of Sciences

ABSTRACT

This paper describes the recent development of ESPnet (https://github.com/espnet/espnet), an end-to-end speech processing toolkit. This project was initiated in December 2017 to mainly deal with end-to-end speech recognition experiments based on sequence-to-sequence modeling. The project has grown rapidly and now covers a wide range of speech processing applications. Now ESPnet also includes text to speech (TTS), voice conversation (VC), speech translation (ST), and speech enhancement (SE) with support for beamforming, speech separation, denoising, and dereverberation. All applications are trained in an end-to-end manner, thanks to the generic sequence to sequence modeling properties, and they can be further integrated and jointly optimized. Also, ESPnet provides reproducible all-in-one recipes for these applications with state-of-the-art performance in various benchmarks by incorporating transformer, advanced data augmentation, and conformer. This project aims to provide up-to-date speech processing experience to the community so that researchers in academia and various industry scales can develop their technologies collaboratively.

Index Terms— End-to-end neural network, speech recognition, text-to-speech, speech translation, speech enhancement

1. INTRODUCTION

The rapid growth of deep learning techniques has made significant changes and improvements in various speech processing algorithms. Automatic speech recognition (ASR) is one of the successful examples in this trend, which achieved significant performance gains with a hybrid model based on hidden Markov model (HMM) and deep neural network (DNN) [1]. The emergent sequence to sequence techniques further accelerate this trend [2, 3]. Thus, we realized end-to-end neural ASR modeling based on these sequence to sequence techniques [4, 5, 6].

Due to the significant demand to establish end-to-end ASR and other speech processing applications, we started developing ESPnet, an end-to-end speech processing toolkit, in December 2017. Our original implementation followed the success of Kaldi speech recognition toolkit [7] and leveraged deep learning frameworks based on Chainer [8] and PyTorch [9]. This paper introduces the recent advances of the ESPnet project since our previous official publication of the whole ESPnet project in 2018 [10].

1.1. Related framework

There are a number of excellent deep learning frameworks that realize similar functions to what ESPnet covers, e.g., Fairseq [11], RETURNN [12], Lingvo [13], and NeMo [14]. These frameworks provide many AI applications, including various natural language processing (NLP) and speech processing methods, based on sequence-to-sequence modeling. Most frameworks include ASR, Text-to-Speech (TTS), and neural machine translation or speech translation (ST). Compared with them, ESPnet focuses more on a wide range of speech applications, and in addition to the above applications, ESPnet also supports various speech enhancement functions now. Also, there are numbers of specialized toolkits for each speech application including Wav2Letter [15] and Espresso [16] for ASR, Fairseq-S2T [17] for ST, and Asteroid [18] for speech separation. The above activities are complementary, and we are closely collaborating/interacting with them, especially with Facebook torchaudio, Nivida Nemo, and Asteroid teams.

1.2. Development history since 2018

Broadened applications The most remarkable update in the recent advances of ESPnet is to broaden the applications from ASR tasks. TTS and Voice conversion (VC) are the most notable extensions, which will be discussed in Section 3 as ESPnet-TTS. Note that ESPnet-TTS is also designed to integrate with ASR/TTS joint training [19] as one of the core research topics in the Fifth Frederick Jelinek Memorial Summer Workshop. Speech translation was also developed to support our research activities to simplify complex speech translation studies with reproducible all-in-one recipes, which will be discussed in Section 4. Section 5 describes one of the most recent significant updates in terms of the applications by involving speech enhancement, which includes beamforming, speech separation, denoising, and dereverberation. In addition to these major speech applications, ESPnet also include CTC-based voice activity detection [20] and CTC-based force alignment [21].

Notable neural architectures and learning methods ESPnet has promptly followed various techniques to provide state-of-the-art results for the community. For example, we have been developing the several ASR decoding algorithms, as described in Sections 2.1, 2.2, and 2.6, which aim to realize fast, unified, and parallelizable inference algorithms. Transformer, conformer, and data augmentation techniques, as described in Sections 2.3 and 2.4 significantly im-

prove the ASR performance, and they are also used in other ESPnet applications. Section 2.5 also describes RNN-Transducer, which is another crucial example aimed for on-line/streaming ASR.

New DNN training system based on ESPnet2 In the release of ESPnet v.0.7.0, we created a new system for DNN training to extend our system for future developments. We referred this project as ESPnet2. Although the main purpose of ESPnet2 is refactoring, it is not limited to refining the source code only. We added several new/enhanced features, including distributed training, on-the-fly feature extraction from the raw waveform, and solving a memory allocation issue of original ESPnet when training on a large scale corpus. The training system of ESPnet2 is shared with all DNN tasks, ASR, TTS, SE, etc., and we can easily integrate a new task with this system. We have already migrated ASR and TTS parts to ESPnet2 and also planning to migrate all existing features in the next steps.

In addition to the above applications, techniques, and ESPnet2-based new training system, we have also improved software workflow by enhancing the continuous integration, enriching documentation, supporting the docker, pip install, and model zoo functions.

1.3. Activity statistics

Table 1 discusses the activity statistics about ESPnet between August 2018 (when [10] was presented at Interspeech 2018), December 2019, and the current one (December 2020). We obtained the previous GitHub statistics from `web.archive.org`[1]. The citation counts were obtained from Google Scholar, accessed on December 10, 2020. Table 1 shows that all GitHub development statistics and citation counts have significantly increased in these two years. In terms of the GitHub development statistics, the large increase in the numbers of contributors, forks, and commits shows that many active developers have increasingly supported the development of ESPnet. The number of citations has also been growing from 19 in 2018 to 95 in 2019, and 205 in 2020. This citation count shows that ESPnet has been used in various research groups and contributed a lot to speech research activities. Finally, we also list the numbers of supported recipes in these periods. We also started to release the ESPnet2 recipes on May 24, 2020, and it includes 27 recipes already.

2. ESPNET-ASR

2.1. Advanced decoding

The basic decoding algorithm in ESPnet-ASR follows an output-label synchronous beam search, which efficiently finds the most probable label sequence for a given input utterance using an encoder-decoder (RNN, Transformer, etc.), CTC and a language model (LM) [22]. CTC guides the beam search process to keep valid hypotheses with monotonic alignments, excluding those with irrelevant alignments. The LM provides a significant improvement of accuracy when it is trained with a large amount of in-domain text.

The decoding module also includes the following features:

Accelerated decoding ESPnet accelerates the decoding process by vectorizing multiple hypotheses during the beam search [23], where the score accumulation steps for each hypothesis are implemented as vector-matrix operations for the vectorized hypotheses. This strategy allows us to take advantage of the parallel computing capabilities of multi-core CPUs and GPUs, resulting in significant
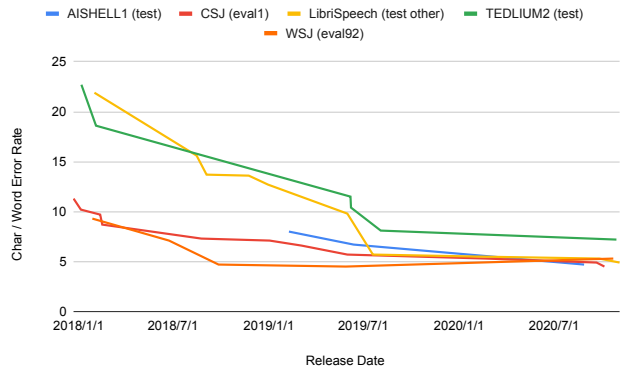

Fig. 1. A history of char/word error rates (CER/WER) on various ASR tasks in ESPnet.

speedups. This manner is also applied to the scoring steps by the CTC and LM to reduce the overhead.

Use of word-based LM ESPnet typically employs an encoder-decoder and an LM relying on a common label set for the sake of simplicity in decoding [22], but it also allows us to use such models while relying on different label sets. For example, we can combine a word-based LM on top of the encoder-decoder that emits letters for English ASR. This kind of configuration is useful for the case that a sufficient amount of paired data is not available to train an encoder-decoder that emits labels longer than letters, and a large amount of external text is available to reliably train the word-based LM. Currently, ESPnet supports multi-level LM [24] and look-ahead LM [25] as a framework to incorporate a word-based LM into the decoding process.

2.2. Generalized decoding

Recent ESPnet models inherit an abstract class named `ScorerInterface` for model-agnostic beam search decoding. It receives encoder output sequences, hypotheses, and internal states (i.e., cache) as inputs, and returns scores of decoder outputs to append to the hypotheses. Most of the models, for example, RNN ASR models, and LM implement this unified interface. In decoding, we combine them as one PyTorch module that can easily run in CPUs and GPUs as well as fairseq [11]'s search module[2]. Similar to Lingvo [13]'s beam search[3], we support efficient joint scoring [26] with CTC and n-gram LM (e.g., KenLM [27]) by a specialized class named `PartialScorerInterface`. It scores a subset of hypotheses with higher scores in the preceded faster models.

2.3. Transformer

Figure 1 illustrates a history of reported char/word error rates (CER/WER) on major ASR corpora in ESPnet. In Figure 1, we can observe some large reductions in 2019 by Transformer-based ASR and data augmentation [28]. The Transformer [29] is a feed-forward architecture using self-attention mechanism for sequence modeling. One drawback in the Transformer is slow decoding speed because it transforms an entire sequence to decode every single token, while the RNN requires just one recurrent frame. To address this problem,

---

	Citations	Contributors	Watches	Stars	Forks	Commits	Issues/PRs	Recipes
Aug. 2018 (v.0.2.0)	19	17	57	610	141	1,216	380	19
Dec. 2019 (v.0.6.0)	95	54	114	1.7K	506	5,994	1,484	47
Dec. 2020 (v.0.9.6)	321	94	160	3.1K	976	10,080	2,768	62 (27)

Table 1. Activity statistics, including GitHub development statistics, citation counts, and the numbers of supported recipes. The numbers of recipes in December 2020 include 62 recipes in ESPnet1 and 27 recipes in ESPnet2.

the `ScorerInterface` caches its previous attention matrix as a state, and computes only new outer vectors to append to the matrix for each step. This method accelerates not only our Transformer-based ASR but also TTS inference as fast as RNN.

2.4. Conformer

In Figure 1, we can observe further improvement in 2020 by Conformer-based ASR [30]. While Transformer models can learn long-range global context better than RNN models, they are less capable to exploit the local information. To address this drawback, Gulati et al. [31] proposed to combine both self-attention and convolution in the encoder, which is named Conformer encoder. With this design, the self-attention module captures the global context while the convolution module exploits the local correlations synchronously. By integrating the Conformer encoder with a Transformer decoder, our models achieved WER of 4.9% on the LibriSpeech test_other task, CER of 4.7% on the AISHELL1 test task. Besides, we also obtained a 7% relative improvement on the multi-speaker WSJ-2mix data and a more than 15% relative improvement on 8 low-resource language corpora compared with Transformer models [30].

2.5. RNN-Transducer

Alongside CTC, attention and hybrid models, ESPnet also supports models based on the RNN-T loss proposed by A. Graves [32]. Following recent advances, different encoder-decoder architecture are also available for these models such as RNN, Transformer (2.3) and Conformer (2.4), but also *free-form* architecture. The latter, exclusive to transducer-based models in ESPnet, allows previously described neural networks and additional ones (TDNN and CausalConv1d [33]) to be combined freely to form a *customized* encoder and decoder architecture definition[4]. For inference with transducer-based models, we follow a different decoding procedure than 2.1 but share the interface described in 2.2. Here, we propose four decoding algorithms allowing more flexibility in regards to the performance-speed trade-off: greedy decoding constrained to one expansion step, beam search [32] (without prefix search), time-synchronous decoding [34], alignment-length decoding [34] and n-step constrained beam search with our modified version of one-step constrained beam search [35]. Shallow fusions with RNN-based and Transformer-based LM and multi-level LM decoding are also supported (2.1).

At the time of writing, we made available in ESPnet various training and decoding configurations and report results for a few corpora: VIVOS (Vietnamese), Voxforge (Italian), Commonvoice (Czech and Welsh) and plan to expand the coverage to other corpora in the following months. Additionally, transducer-based models from ESPnet were reported successful by various researchers with other corpora and languages: OpenSTT (Russian) [36], ESTER (French) [37] and CHiME-6 Challenge [38].

2.6. Non-autoregressive modeling

In addition to the autoregressive (AR) modeling based on an encoder-decoder architecture [5, 6], ESPnet supports non-autoregressive (NAR) modeling of end-to-end ASR models. AR models suffer from slow inference speed, which requires as many forward calculations of the decoder as the length of an output sequence. On the other hand, NAR models permit for fast sequence generation with a constant number of the decoder calculations. Recently proposed Mask-CTC [39] realizes the NAR modeling based on mask-predict [40] and CTC. During inference, a target sequence is generated by iteratively refining an output of CTC with mask-predict. Mask-CTC achieves fast inference time (< 0.1 RTF using a single CPU) and competitive performance to that of the AR model using Conformer (Section 2.4) [41].

3. ESPNET-TTS

3.1. TTS

As an extension of ESPnet-ASR, we released ESPnet-TTS that supports text-to-speech (TTS) task [42], which mainly focuses on the development of text to mel-spectrogram (text2mel) models. Initially, ESPnet-TTS has been developed with ESPnet1, but recently it is also available in ESPnet2. It supports AR text2mel models such as Tacotron 2 [43] and Transformer-TTS [44], NAR models such as FastSpeech [45] and FastSpeech 2 [46], and their multi-speaker extensions with X-vector [47] and global style token [48]. Thanks to the unified libraries with various tasks, it can quickly introduce new architectures from the other tasks into TTS (e.g., Conformer-based FastSpeech). As mel-spectrogram to waveform (mel2wav, i.e., vocoder) models, we support WaveNet vocoder [49], Parallel WaveGAN [50], MelGAN [51], and Multi-band MelGAN [52] with external libraries[5][6]. In addition that users can quickly develop the state-of-the-art baseline systems for the research purpose, they can easily make a demonstration system, which works in real-time for various languages, including English, Mandarin, and Japanese[7].

3.2. VC

Considering that TTS and VC share the same goal of performing speech synthesis, it is easy to extend the TTS models into VC models. We recently released VC recipes that can be trained with a parallel corpus (i.e. pair of sentences with the same content uttered by the source and target speaker) for mel-spectrogram-to-mel-spectrogram conversion. We support Tacotron2-based and Transformer-based models, and we provided a pre-trained model that enables sample efficient training such that only approximately 5 minutes of data is needed [53]. In addition, an any-to-one system that chains an ASR and a TTS model was developed as the baseline system [54] of the voice conversion challenge 2020 [55]. Taking advantage of the well-tuned pre-trained ASR and TTS models provided in ESPnet, this

---

[4]https://espnet.github.io/espnet/tutorial.html#transducer

[5]https://github.com/r9y9/wavenet_vocoder
[6]https://github.com/kan-bayashi/ParallelWaveGAN
[7]The realtime demo is available at https://bit.ly/3a54G3s.

simple yet effective system was placed 2nd in terms of conversion similarity in the official listening test.

4. ESPNET-ST

Although ESPnet has supported ASR and TTS tasks as initial speech applications, we recently started to support the speech translation (ST) task [56]. Both the traditional pipeline approach, where ASR and text-based machine translation (MT) modules are cascaded, and end-to-end (E2E) approach, where source speech is directly translated to text in another language, are readily available in ESPnet-ST. Our goal is to build reliable baselines quickly with minimal effort, and we demonstrated the state-of-the-art translation performance [56]. Moreover, new progresses in ASR can be easily transferred to ST performance improvement since both tasks are seamlessly integrated into a unified ESPnet framework. For instance, we presented that a better encoder architecture which was originally proposed in the ASR task, Conformer (Section 2.4), is effective for the E2E-ST task as well in [30]. Multilinguality is also a good example in the ST task [57]. To speed up inference, non-autoregressive (NAR) E2E-ST models have been studied on ESPnet-ST recently [58], similar to NAR ASR in Section 2.6. We will officially support various NAR methods in the future.

5. ESPNET-SE

ESPnet-SE [59] is a recently released component of ESPnet2. It has the capability to process various speech enhancement tasks, including speech dereverberation, denoising and speech separation for both single-channel and multi-channel data. ESPnet-SE is designed to be easily integrated with the downstream ASR tasks, including the robust ASR and multi-talker ASR. The enhancement front-end can be also jointly trained with ASR tasks.

5.1. Single-channel enhancement/separation

In the single-channel condition, we have implemented various popular models for speech enhancement and separation. Both frequency-domain [60, 61] models and time-domain [62, 63] models are supported. The networks and loss functions used in the speech enhancement/separation are flexible with configuration files, thus the newest models can be quickly reproduced and included in the toolkit. The input and output of the SE model interface are raw waveforms, which makes it easier for the integration with other downstream tasks. Depending on the number of speech sources, single-talker speech enhancement and multi-talker speech separation are designed into a unified framework in ESPnet-SE.

5.2. Multi-channel enhancement/separation

When processing multi-channel speech data collected by multiple microphones or a microphone array, the additional spatial information can be exploited to achieve better performance with ESPnet-SE. The current model `BeamformerNet` is mainly based on neural beamformers [64, 65], with support for various beamformer types including minimum variance distortionless response (MVDR) [66], weighted power minimization distortionless response (WPD) [67], and weighted minimum power distortionless response (wMPDR) [68]. It is worth noting that `BeamformerNet` is implemented in a fully differentiable way, thus allowing end-to-end optimization with either signal-level objectives or the ASR criterion.

5.3. Dereverberation

Support for DNN based weighted prediction error (WPE) dereverberation [69] is added [8]. The dereverberation network can be optimized using the target signal like [69]. We also provide an option to connect it as a dereverberation subnetwork before the differential beamformer described in Sec. 5.2 and train it end-to-end with just speech recognition objectives as used in [70]. Evaluation is performed with all the speech enhancement metrics prescribed in the REVERB challenge [71] in addition to the ASR metrics.

5.4. Multi-speaker ASR

In addition to the multi-channel speech separation in Sec. 5.2, the end-to-end multi-channel multi-speaker speech recognition [72, 73] is supported in ESPnet-SE. In line with the multi-channel speech separation, the input is the multi-channel speech signals containing overlapped speech from multiple speakers. The model can be divided into two modules, including speech separation and recognition. Firstly, the speech separation module is a masking network-based neural beamformer as introduced in Sec. 5.2. The masking network first separates the overlapped speech by predicting signal masks for each speaker and an additional noise mask over all channels. Then the beamformer enhances the speech of each speaker. Secondly, the speech recognition module, based on the joint CTC /attention-based E2E ASR framework [22], takes the separated speech of each speaker as input and outputs the corresponding transcriptions. All the parameters in the whole model are optimized simultaneously based on the final ASR loss only. To address the label permutation problem in computing the ASR loss, we compute the ASR loss with permutation invariant training (PIT), which is similar to the method used in previous end-to-end multi-speaker ASR systems [74, 75].

5.5. Joint training

Thanks to the complete modules of both speech enhancement and recognition designed in an unified framework, the joint training of both the front-end model and the back-end model can be achieved without any hassle. Based on the foundation from single/multi-channel enhancement/separation in Sec. 5.1, 5.2, 5.3 and multi-speaker ASR in 5.4, joint-training incorporates the models and functions from both, along with some specific features related to the joint-training, such as the composition of the total loss, the formation of the enhanced sources, the criterion to permute the sources and so on. In general, with a serial pipeline from enhancement to recognition, all the built-in models from Sec. 5.1, 5.2, 5.3 and 5.4 could be configurable, corresponding to the input data. The whole design of joint-training is end-to-end, and all the parameters in the whole model could be optimized with the combination of enhancement loss and ASR loss or just the final ASR loss. In terms of the application, jointly training models avoid the complexity from the individual training of the front-end and back-end, as well as the difficulties to combine these two parts in both training and inference phase. After the experiments on some benchmark datasets in multi-speaker scenes, e.g. WSJ0-2mix [76], the jointly trained models attain performance that is comparable to or even better than the baseline models.

---

[8]DNN-WPE module from `https://github.com/nttcslab-sp/dnn_wpe`

6. SUMMARY AND FUTURE PLANS

This paper introduced the recent advances of the ESPnet end-to-end speech processing toolkit. The ESPnet project has been rapidly growing by covering a wide range of speech applications and incorporating state-of-the-art techniques based on community-driven developments by many supporters. Our future work will further accelerate the project by considering the demand of the community and research trends. For example, we will focus on enhancing online/streaming ASR functions from our current RNN-T implementation and further broadening the speech applications by realizing end-to-end speech-to-speech translation and spoken dialogue systems within the framework. We will also focus on realizing complete speech conversation understanding systems that can recognize who spoke what, when, and where by incorporating models such as [77].

7. REFERENCES

[1] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. EMNLP*, 2014, pp. 1724–1734.

[3] I. Sutskever, O. Vinyals, Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NeurIPS*, 2014, pp. 3104–3112.

[4] A. Graves, A.-r. Mohamed, G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.

[5] J. K. Chorowski *et al.*, "Attention-based models for speech recognition," *Proc. NeurIPS*, vol. 28, pp. 577–585, 2015.

[6] W. Chan, N. Jaitly, Q. Le, O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[7] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[8] S. Tokui, K. Oono, S. Hido, J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proc. LearningSys in NeurIPS*, 2015.

[9] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, 2019, pp. 8024–8035.

[10] S. Watanabe *et al.*, "ESPnet: End-to-end speech processing toolkit," *Proc. Interspeech*, pp. 2207–2211, 2018.

[11] M. Ott *et al.*, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proc. NAACL-HLT: Demonstrations*, 2019.

[12] A. Zeyer, T. Alkhouli, H. Ney, "RETURNN as a generic flexible neural toolkit with application to translation and speech recognition," in *Proc. ACL: System Demonstrations*, 2018, pp. 128–133.

[13] J. Shen *et al.*, "Lingvo: a modular and scalable framework for sequence-to-sequence modeling," *arXiv preprint arXiv:1902.08295*, 2019.

[14] O. Kuchaiev *et al.*, "Nemo: a toolkit for building ai applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.

[15] R. Collobert, C. Puhrsch, G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[16] Y. Wang *et al.*, "Espresso: A fast end-to-end neural speech recognition toolkit," in *Proc. ASRU*, 2019, pp. 136–143.

[17] C. Wang *et al.*, "fairseq s2t: Fast speech-to-text modeling with fairseq," *arXiv preprint arXiv:2010.05171*, 2020.

[18] M. Pariente *et al.*, "Asteroid: the pytorch-based audio source separation toolkit for researchers," *arXiv preprint arXiv:2005.04132*, 2020.

[19] T. Hori *et al.*, "Cycle-consistency training for end-to-end speech recognition," in *Proc. ICASSP*, 2019, pp. 6271–6275.

[20] T. Yoshimura, T. Hayashi, K. Takeda, S. Watanabe, "End-to-end automatic speech recognition integrated with CTC-based voice activity detection," in *Proc. ICASSP*, 2020, pp. 6999–7003.

[21] L. Kürzinger *et al.*, "CTC-segmentation of large corpora for german end-to-end speech recognition," *arXiv preprint arXiv:2007.09127*, 2020.

[22] T. Hori, S. Watanabe, Y. Zhang, W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," *Proc. Interspeech*, pp. 949–953, 2017.

[23] H. Seki *et al.*, "Vectorized beam search for CTC-attention-based speech recognition," *Proc. Interspeech*, pp. 3825–3829, 2019.

[24] T. Hori, S. Watanabe, J. R. Hershey, "Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition," in *Proc. ASRU*, 2017, pp. 287–293.

[25] T. Hori, J. Cho, S. Watanabe, "End-to-end speech recognition with word-based E language models," in *Proc. SLT*, 2018, pp. 389–396.

[26] S. Watanabe *et al.*, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE JSTSP*, vol. 11, no. 8, pp. 1240–1253, 2017.

[27] K. Heafield, I. Pouzyrevsky, J. H. Clark, P. Koehn, "Scalable modified Kneser-Ney language model estimation," in *Proc. ACL: Short Papers*, 2013, pp. 690–696.

[28] S. Karita *et al.*, "A comparative study on Transformer vs RNN in speech applications," in *Proc. ASRU*, 2019, pp. 449–456.

[29] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017, vol. 30, pp. 5998–6008.

[30] P. Guo *et al.*, "Recent developments on ESPnet toolkit boosted by Conformer," *arXiv preprint arXiv:2010.13956*, 2020.

[31] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. INTERSPEECH*, 2020, pp. 5036–5040.

[32] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv:1211.3711 (arXiv preprint)*, 2012.

[33] C. Weng *et al.*, "Minimum bayes risk training of rnn-transducer for end-to-end speech recognition," *arxiv:1911.12487 (arXiv preprint)*, 2019.

[34] G. Saon, Z. Tüske, K. Audhkhasi, "Alignment-length synchronous decoding for RNN transducer," in *Proc. ICASSP*, 2020, pp. 7804–7808.

[35] J. Kim, Y. Lee, "Accelerating rnn transducer inference via one-step constrained beam search," *arXiv:2002.03577 (arXiv preprint)*, 2020.

[36] A. Andrusenko, A. Laptev, I. Medennikov, "Exploration of end-to-end asr for openstt – russian open speech-to-text dataset," in *SPECOM*, 2020, pp. 35–44.

[37] F. Boyer, J.-L. Rouas, "End-to-end speech recognition: A review for the french language," in *arXiv:1910.08502 (arXiv preprint)*, 2019.

[38] A. Andrusenko, A. Laptev, I. Medennikov, "Towards a competitive end-to-end speech rrecognition for chime-6 dinner party transcription," *arxiv:2004.10799 (arXv preprint)*, 2020.

[39] Y. Higuchi *et al.*, "Mask CTC: Non-autoregressive end-to-end ASR with CTC and mask predict," in *Proc. Interspeech*, 2020, pp. 3655–3659.

[40] M. Ghazvininejad, O. Levy, Y. Liu, L. Zettlemoyer, "Mask-predict: Parallel decoding of conditional masked language models," in *Proc. EMNLP-IJCNLP*, 2019, pp. 6114–6123.

[41] Y. Higuchi *et al.*, "Improved Mask-CTC for non-autoregressive end-to-end ASR," *arXiv preprint arXiv:2010.13270*, 2020.

[42] T. Hayashi *et al.*, "Espnet-tts: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *Proc. ICASSP*, 2020, pp. 7654–7658.

[43] J. Shen *et al.*, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[44] N. Li *et al.*, "Close to human quality TTS with Transformer," *arXiv preprint arXiv:1809.08895*, 2018.

[45] Y. Ren *et al.*, "Fastspeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, 2019, pp. 3165–3174.

[46] Y. Ren *et al.*, "FastSpeech 2: Fast and high-quality end-to-end text-to-speech," *arXiv preprint arXiv:2006.04558*, 2020.

[47] D. Snyder *et al.*, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, 2018, pp. 5329–5333.

[48] Y. Wang *et al.*, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[49] A. v. d. Oord *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[50] R. Yamamoto, E. Song, J.-M. Kim, "Parallel waveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. ICASSP*, 2020, pp. 6199–6203.

[51] K. Kumar *et al.*, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Proc. NeurIPS*, 2019, pp. 14910–14921.

[52] G. Yang *et al.*, "Multi-band melGAN: Faster waveform generation for high-quality text-to-speech," *arXiv preprint arXiv:2005.05106*, 2020.

[53] W.-C. Huang *et al.*, "Voice Transformer Network: Sequence-to-Sequence Voice Conversion Using Transformer with Text-to-Speech Pretraining," in *Proc. Interspeech*, 2020, pp. 4676–4680.

[54] W.-C. Huang, T. Hayashi, S. Watanabe, T. Toda, "The Sequence-to-Sequence Baseline for the Voice Conversion Challenge 2020: Cascading ASR and TTS," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 160–164.

[55] Z. Yi *et al.*, "Voice Conversion Challenge 2020 — Intra-lingual semi-parallel and cross-lingual voice conversion —," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 80–98.

[56] H. Inaguma *et al.*, "ESPnet-ST: All-in-one speech translation toolkit," in *Proc. ACL: System Demonstrations*, 2020, pp. 302–311.

[57] H. Inaguma, K. Duh, T. Kawahara, S. Watanabe, "Multilingual end-to-end speech translation," in *Proc. ASRU*, 2019, pp. 570–577.

[58] H. Inaguma *et al.*, "Orthros: Non-autoregressive end-to-end speech translation with dual-decoder," *arXiv preprint arXiv:2010.13047*, 2020.

[59] C. Li *et al.*, "ESPnet-SE: end-to-end speech enhancement and separation toolkit designed for ASR integration," *arXiv preprint arXiv:2011.03706*, 2020.

[60] D. Yu, M. Kolbæk, Z.-H. Tan, J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. ICASSP*, 2017, pp. 241–245.

[61] M. Kolbæk, D. Yu, Z.-H. Tan, J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Trans. ASLP*, vol. 25, no. 10, pp. 1901–1913, 2017.

[62] Y. Luo, N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *Proc. ICASSP*, 2018, pp. 696–700.

[63] Y. Luo, N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM trans. ASLP*, vol. 27, no. 8, pp. 1256–1266, 2019.

[64] J. Heymann, L. Drude, R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. ICASSP*, 2016, pp. 196–200.

[65] H. Erdogan *et al.*, "Improved MVDR beamforming using single-channel mask prediction networks," in *Proc. Interspeech*, 2016, pp. 1981–1985.

[66] B. D. Van Veen, K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[67] T. Nakatani, K. Kinoshita, "A unified convolutional beamformer for simultaneous denoising and dereverberation," *IEEE Signal Processing Letters*, vol. 26, no. 6, pp. 903–907, 2019.

[68] C. Boeddeker, T. Nakatani, K. Kinoshita, R. Haeb-Umbach, "Jointly optimal dereverberation and beamforming," in *Proc. ICASSP*, 2020, pp. 216–220.

[69] K. Kinoshita et al., "Neural network-based spectrum estimation for online WPE dereverberation," in *Proc. Interspeech*, 2017, pp. 384–388.

[70] A. S. Subramanian *et al.*, "Speech enhancement using end-to-end speech recognition objectives," in *WASPAA*, 2019, pp. 229–233.

[71] K. Kinoshita *et al.*, "The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proc. WASPAA*, Oct 2013, pp. 1–4.

[72] X. Chang *et al.*, "MIMO-Speech: End-to-end multi-channel multi-speaker speech recognition," in *Proc. ASRU*, 2019, pp. 237–244.

[73] X. Chang *et al.*, "End-to-end multi-speaker speech recognition with Transformer," in *Proc. ICASSP*, 2020, pp. 6134–6138.

[74] H. Seki *et al.*, "A purely end-to-end system for multi-speaker speech recognition," in *Proc. ACL: Long Papers*, 2018, pp. 2620–2630.

[75] X. Chang, Y. Qian, K. Yu, S. Watanabe, "End-to-end monaural multi-speaker ASR system without pretraining," in *Proc. ICASSP*, 2019, pp. 6256–6260.

[76] J. R. Hershey, Z. Chen, J. L. Roux, S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, 2016.

[77] A. S. Subramanian *et al.*, "Directional ASR: A new paradigm for E2E multi-speaker speech recognition with source localization," 2020.