INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

FunASR: A Fundamental End-to-End Speech Recognition Toolkit

Zhifu Gao, Zerui Li, Jiaming Wang, Haoneng Luo, Xian Shi, Mengzhe Chen, Yabin Li, Lingyun Zuo, Zhihao Du, Shiliang Zhang

Speech Lab of DAMO Academy, Alibaba Group, China

{zhifu.gzf, lzr265946, wangjiaming.wjm, haoneng.lhn, shixian.shi, mengzhe.cmz, wucong.lyb, ailsa.zly, neo.dzh, sly.zsl}@alibaba-inc.com

Abstract

This paper introduces FunASR[1], an open-source speech recognition toolkit designed to bridge the gap between academic research and industrial applications. FunASR offers models trained on large-scale industrial corpora and the ability to deploy them in applications. The toolkit's flagship model, Paraformer, is a non-autoregressive end-to-end speech recognition model that has been trained on a manually annotated Mandarin speech recognition dataset that contains 60,000 hours of speech. To improve the performance of Paraformer, we have added timestamp prediction and hotword customization capabilities to the standard Paraformer backbone. In addition, to facilitate model deployment, we have open-sourced a voice activity detection model based on the Feedforward Sequential Memory Network (FSMN-VAD) and a text post-processing punctuation model based on the controllable time-delay Transformer (CT-Transformer), both of which were trained on industrial corpora. These functional modules provide a solid foundation for building high-precision long audio speech recognition services. Compared to other models trained on open datasets, Paraformer demonstrates superior performance.

Index Terms: FunASR, Paraformer, Speech Recognition, FSMN-VAD, CT-Transformer

1. Introduction

Over the past few years, the performance of end-to-end (E2E) models has surpassed that of conventional hybrid systems on automatic speech recognition (ASR) tasks. There are three popular E2E approaches: connectionist temporal classification (CTC) [1], recurrent neural network transducer (RNN-T) [2] and attention based encoder-decoder (AED) [3, 4]. Of these, AED models have dominated seq2seq modeling for ASR, due to their superior recognition accuracy [4–13]. Open-source toolkits including ESPNET [14], WeNet [15], PaddleSpeech [16] and K2 [17] et al., have been developed to facilitate research in end-to-end speech recognition. These open-source tools have played a great role in reducing the difficulty of building an end-to-end speech recognition system.

In this work, we introduce FunASR, a new open source speech recognition toolkit designed to bridge the gap between academic research and industrial applications. FunASR builds upon previous works and provides several unique features:

1. *Modelsope*: FunASR provides a comprehensive range of pre-trained models based on industrial data. The flagship model, Paraformer [18], is a non-autoregressive end-to-end speech recognition model that has been trained on a manually annotated Mandarin speech recognition dataset that contains 60,000 hours of speech. Compared with Conformer [5] and RNN-T [2] supported by mainstream open source frameworks, Paraformer offers comparable performance while being more efficient.
2. *Training & Finetuning*: FunASR is a comprehensive toolkit that offers a range of example recipes to train end-to-end speech recognition models from scratch, including Transformer, Conformer, and Paraformer models for datasets like AISHELL [19, 20], WenetSpeech [21] and LibriSpeech [22]. Additionally, FunASR provides a convenient finetuning script that allows users to quickly fine-tune a pre-trained model from the ModelScope on a small amount of domain data, resulting in high-performance recognition models. This feature is particularly beneficial for academic researchers and developers who may have limited access to data and computing power required to train models from scratch.
3. *Speech Recognition Services*: FunASR enables users to build speech recognition services that can be deployed on real-applications. To facilitate model deployment, we have also released a voice activity detection model based on the Feedforward Sequential Memory Network (FSMN-VAD) [23] and a text post-processing punctuation model based on the controllable time-delay Transformer (CT-Transformer) [24], both of which were trained on industrial corpora. To improve the performance of Paraformer, we have added timestamp prediction and hotword customization capabilities to the standard Paraformer backbone. Additionally, FunASR includes an inference engine that supports CPU and GPU inference through ONNX, libtorch, and TensorRT. These functional modules simplify the process of building high-precision, long audio speech recognition services using FunASR.

[1]https://github.com/alibaba-damo-academy/FunASR

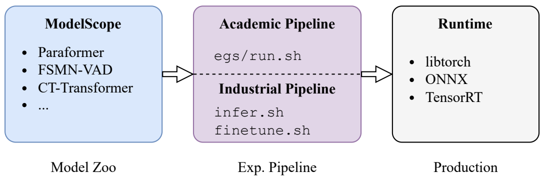

Figure 1: *Overview of FunASR design.*

Overall, FunASR is a powerful speech recognition toolkit that offers unique features not found in other open source tools. We believe that our contributions will help to further advance the field of speech recognition and enable more researchers and developers to apply these techniques to real-world applications. It should be noted that, this paper only reports the experiments on Mandarin corpora, due to the limitation of the number of pages. In fact, FunASR supports many types of languages, including English, French, German, Spanish, Russian, Japanese, Korean, etc (more details could be found in model zoo).

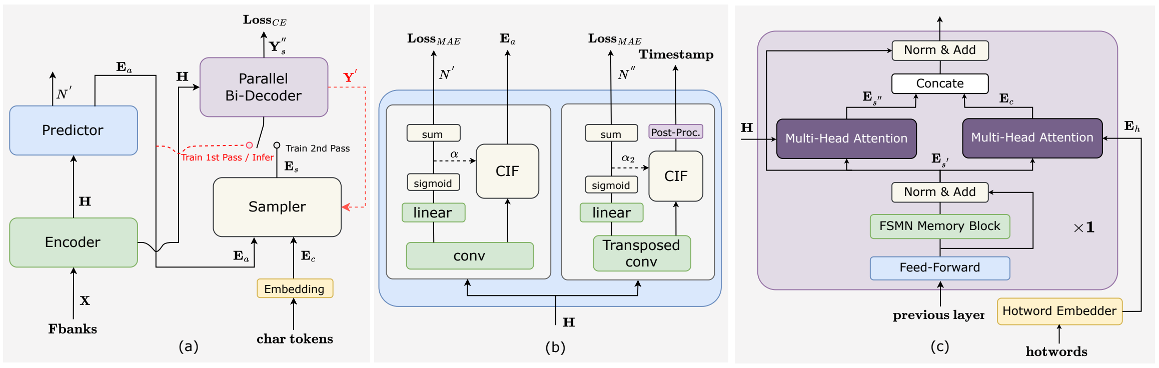

Figure 2: *Illustrations of the Paraformer related architectures. (a) Paraformer; (b) Advanced timestamp prediction; (c) Contextual decoder layer for hotword customization.*

2. Overview of FunASR

The overall framework of FunASR is illustrated in Fig. 1. ModelScope manages the models utilized in FunASR and hosts critical ones such as Paraformer, FSMN-VAD, and CT-Transformer.

Users of FunASR can easily perform experiments using its Pytorch-based pipelines, which are categorized as either academic or industrial pipelines. The academic pipeline, denoted by $run.sh$, enables users to train models from scratch. The $run.sh$ script follows the recipe style of ESPNET and includes stages for data preparation (stage 0), feature extraction (stage 1), dictionary generation (stage 2), model training (stage 3 and 4), and model inference and scoring (stage 5). In contrast, the industrial pipeline offers two separate scripts: $infer.sh$ for inference and $finetune.sh$ for fine-tuning. These pipelines are easy to use, with users only needing to specify the model name and dataset.

FunASR also provides an easy-to-use runtime for deploying models in applications. To support various hardware platforms such as CPU, GPU, Android, and iOS, we offer different runtime backends including Libtorch, ONNX, and TensorRT. In addition, we utilize AMP quantization [25] to accelerate the inference runtime and ensure optimal performance. With these features, FunASR makes it easy to deploy and use speech recognition models in a wide range of applications.

3. Main Modules of FunASR

3.1. Paraformer

To begin, let us provide a brief overview of Paraformer [18], a model that we have previously proposed, depicted in Fig. 2(a). Paraformer is a single-step non-autoregressive (NAR) model that incorporates a glancing language model-based sampler module to enhance the NAR decoder's ability to capture token inter-dependencies.

The Paraformer consists of two core modules: the predictor and the sampler. The predictor module is used to generate acoustic embeddings, which capture the information from the input speech signals. During training, the sampler module incorporates target embeddings by randomly substituting tokens into the acoustic embeddings to generate semantic embeddings. This approach allows the model to capture the interdependence between different tokens and improves the overall performance of the model. However, during inference, the sampler module is inactive, and the acoustic embeddings are used to output the final prediction over only a single pass. This approach ensures faster inference times and lower latency.

In order to further enhance the performance of Paraformer, this paper proposes modifications including timestamp prediction and hotword customization. In addition, the loss function used in [18] has been updated by removing the MWER loss, which was found to contribute little to performance gains. An additional CE loss is now used in the first pass decoder to reduce the discrepancy between training and inference. The next subsection will provide detailed explanations.

3.2. Timestamp Predictor

Accurate timestamp prediction is a crucial function of ASR systems. However, conventional industrial ASR systems require an extra hybrid model to conduct force-alignment (FA) for timestamp prediction (TP), leading to increased computation and time costs. FunASR provides an end-to-end ASR model that achieves accurate timestamp prediction by redesigning the structure of the Paraformer predictor, as depicted in Fig.2(b). We introduce a transposed convolution layer and LSTM layer to upsample the encoder output, and timestamps are generated by post-processing CIF [26] weights α_2. We treat the frames between two fireplaces as the duration of the former tokens and mark out the silence parts according to α_2. In addition, FunASR also releases a *force-alignment-like* model named TP-Aligner, which includes an encoder of smaller size and a timestamp predictor. It takes speech and corresponding transcription as input to generate timestamps.

Table 1: *Evaluation of timestamp prediction.*

Data	System	AAS (ms)
AISHELL	Force-alignment	80.1
	Paraformer-TP	71.0
Industrial Data	Force-alignment	60.3
	Paraformer-large-TP	65.3
	TP-Aligner	69.3

We conducted experiments on AISHELL and 60,000-hour industrial data to evaluate the quality of timestamp prediction. The evaluation metrics used for measuring timestamp quality is the accumulated average shift (AAS) [27]. We used a test set of 5,549 utterances with manually marked timestamps to compare the timestamp prediction performance of the provided models with FA systems trained with Kaldi [28]. The results show that Paraformer-TP outperforms the FA system on AISHELL. In industrial experiments, we found that the proposed timestamp prediction method is comparable to the hybrid FA system in terms of timestamp accuracy (with a gap of less than 10ms). Moreover, the one-pass solution is valuable for commercial usage as it helps in reducing computation and time overhead.

3.3. Hotword Customization

Contextual Paraformer offers the ability to customize hotwords by utilizing named entities, which enhances incentives and improves the recall and accuracy. Two additional modules have been added to the basic Paraformer model - a hotword embedder and a multi-head attention in the last layer of the decoder, depicted in Fig. 2(c).

We utilize hotwords, denoted as $\boldsymbol{w} = \boldsymbol{w_1}, ..., \boldsymbol{w_n}$, as input to our hotword embedder [29]. The hotword embedder consists of an embedding layer and LSTM layer, which takes the context hotwords as input and generates an embedding, denoted as $\boldsymbol{E_h}$, by using the last state of the LSTM. Specifically, the hotwords are first fed to the hotword embedder, which produces a sequence of hidden states. We then use the last hidden state as the embedding of the hotwords, capturing the contextual information of the input sequence.

To capture the relationship between the hotword embedding $\boldsymbol{E_h}$ and the output of the last layer of the FSMN memory block $\boldsymbol{E_{s'}}$, we employ a multi-head attention module. Then, we concate the $\boldsymbol{E_{s'}}$ and contextual attention $\boldsymbol{E_c}$. This operation is formalized in Equation 1:

$$\begin{aligned} E_c &= \text{MultiHeadAttention}(E_{s'} W_c^Q, E_h W_c^K, E_h W_c^V), \\ E_{s''} &= \text{MultiHeadAttention}(E_{s'} W_s^Q, H W_s^K, H W_s^V), \quad (1) \\ O &= \text{Conv1d}([E_{s''}; E_c]) \end{aligned}$$

We use a one-dimensional convolutional layer ($Conv1d$) to reduce its dimensionality to match that of the hidden state $\boldsymbol{E_{s'}}$, which serves as the input of the subsequent layer. It's worth noting that apart from this modification, the other processes of our Contextual Paraformer are the same as those of the standard Paraformer.

Table 2: *The test sets used in this customization task.*

Dataset	Utts	Named Entities
AI domain	486	204
Common domain	1308	231

During the training, the hotwords are randomly generated from target in each training batch. As for inference, we can specify hotwords by providing a list of named entities to the model.

Table 3: *Evaluation of hotword customization*

Dataset	Hotwords	CER	R	P	F_1
AISHELL hotword subtest	w/o	10.01	16	100	27
	w/	4.55	74	100	85
Industrial AI domian	w/o	7.96	70	98	82
	w/	6.31	89	98	93
Industrial common domian	w/o	9.47	67	100	80
	w/	8.75	80	98	88

To evaluate the hotword customization effect of Contextual Paraformer, we created a hotword testset by sampling 235 audio clips containing entity words from the AISHELL testset, which included 187 named entities. The dataset has been uploaded to the ModelScope and the test recipe has been opened to FunASR. Additionally, we expanded our experiments to include the AI domain and Common domain of industrial tasks, as presented in Tab. 2.

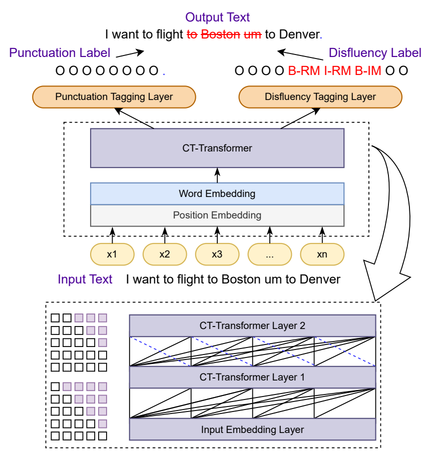

Figure 3: *Illustration of CT-Transformer architecture.*

Tab. 3 presents the experimental results of our study on the impact of hotwords on the performance of Contextual Paraformer. We employed CER and F1-score as evaluation metrics for the customization task. Our results indicate an impressive improvement of approximately 58% in F1-score on the AISHELL-1 named entity subtest. Moreover, we achieved an average of 10% improvement on industrial customization tasks.

3.4. Voice Activity Detection

Voice activity detection (VAD) plays a important role in speech recognition systems by detecting the beginning and end of effective speech. FunASR provides an efficient VAD model based on the FSMN structure [23]. To improve model discrimination, we use monophones as modeling units, given the relatively rich speech information. During inference, the VAD system requires post-processing for improved robustness, including operations such as threshold settings and sliding windows.

Table 4: *Evaluation of VAD on continuous utterances.*

Dataset	VAD	CER	Speech%
Meeting domain	w/o	2.42	1
	w/	2.31	0.92
Video domain	w/o	27.87	1
	w/	25.34	0.78

The evaluation of VAD is presented in detail in Table 4. The test set consists of manually annotated data from two domains: 2 hours of meeting data and 4 hours of video data. We report the Character Error Rate (CER) and the percentage of utterances sent to ASR inference for recognition. The results demonstrate that the VAD effectively filters out invalid voice, allowing the recognition system to focus on effective speech and leading to a significant CER improvement.

3.5. Text Postprocessing

Text postprocessing is a critical step in generating readable ASR transcripts, which involves adding punctuation marks and removing speech disfluencies. FunASR includes a CT-Transformer model that performs both tasks in real-time, as described in [24]. The model's overall framework is presented in Fig. 3. To meet real-time constraints, the model allows partial outputs to be frozen with controllable time delay. A fast decoding strategy is utilized to minimize latency while maintaining

Table 5: *Comparison of CERs on AISHELL, AISHELL-2 and WenetSpeech with open source speech toolkits. RTF is evaluated with batchsize 1.(† : The number of the model parameters is 110M)*

Model	#Parameters	AR/NAR	LM	AISHELL test	AISHELL-2 test_ios	WenetSpeech test_meeting	RTF
ESPNET Conformer [14]	46.25M	AR	w/	4.60	5.70	15.90 †	-
WeNet Conformer-U2++ [30]	47.30M	AR	w/	4.40	5.35	17.34 †	-
PaddleSpeech DeepSpeech2 [16]	58.4M	NAR	w/	6.40	-	-	-
K2 Transducer [17]	80M	AR	w/o	5.05	5.56	14.44 †	-
Conformer	46.25M	AR	w/	4.65	5.35	15.21	1.4300
			w/o	5.21	5.83	-	0.2100
Paraformer	46.3M	NAR	w/o	4.95	5.73	-	0.0168
Paraformer-large	220M	NAR	w/o	1.95	2.85	6.97	0.0251

competitive performance. Moreover, to reduce computational complexity, the strategy dynamically discards a history that is too long based on already predicted punctuation marks.

Table 6: *The results of text postprocessing*

Model	Punctuation			Disfluency			Inference Time
	P	R	F_1	P	R	F_1	
BLSTM	60.2	48.8	53.9	84.1	57.0	67.9	1112.9s (×1.0)
Full-Trans.	62.1	55.9	58.8	83.1	61.2	70.5	676.7s (×1.6)
CT-Trans.	62.7	55.3	58.8	82.4	61.5	70.5	585.8s (×1.9)

We evaluated the punctuation prediction and disfluency detection on an in-house Chinese dataset that consists of about 24K spoken utterances with punctuation and disfluency annotations. Tab. 6 shows the precision (P), recall (R), and F_1-score (F_1) of the models. The results indicate that the CT-Transformer achieves competitive F_1 with a faster inference speed.

4. Experiments

4.1. Evaluation ASR

In our experiments, we evaluate the performance of our models on the AISHELL, AISHELL-2, and WenetSpeech datasets and present the results in detail in Table 5. When compared to other open source toolkits, FunASR achieves comparable results to the baseline Conformer model. To ensure a fair comparison with Paraformer, we remove the language models (LM) and joint-CTC decoding of Conformer. The results show that Paraformer slightly outperforms Conformer without LM in terms of recognition accuracy. We also evaluate the computational efficiency of the models in terms of inference speed by RTF on GPU (V100) with a batch size of 1. The Paraformer model achieves a 12x speedup in inference, which is a significant advantage over autoregressive models. Even when the number of parameters in the Paraformer-large model is increased from 46M to 220M, the RTF still outperforms the AR model with 46M parameters.

FunASR provides a pre-trained Paraformer-large model that is specifically trained on a 60,000 hour Mandarin speech recognition corpus used in industry. The performance of Paraformer-large is impressive, as shown in Table 5. It achieves low CERs of 1.95%, 2.85%, and 6.97% on the AISHELL test, AISHELL-2 test_ios, and WenetSpeech test_meeting task, respectively. These results demonstrate the importance of using large-scale speech corpora in improving the performance of ASR systems.

Moreover, FunASR offers customization capabilities for our pretrain model, allowing it to be finetuned on domain-specific data. The results of using the AISHELL and industrial domain data (200h) to finetune our Paraformer-large model are presented in Table 7. Our model achieved relative improvements of 7.4% and 8.7% on the AISHELL dev and test tasks, respectively. Additionally, the experiments in the logistics field have demonstrated that finetuning the Paraformer-large model results in a significant improvement in recognition and domain keywords recall. The short testset consists of audio files that are 5.77s long on average, with a total duration of 2.41 hours. The long testset is 4.95 hours long and contains audio with an average length of 162.0s. On average, we observed an increase in domain keywords recall from 76.7% to 96.8%, and a decrease in CER from 11.25% to 10.10%. The experiments have demonstrated that the pretrained model can be finetuned with your corpus to achieve significant improvements in the relevant field.

Table 7: *Evaluation of funetuning*

Model	Dataset	Pretrain(CER%/Recall%)	Finetune(CER%/Recall%)
AISHELL	dev	1.75 / -	1.62 / -
	test	1.95 / -	1.78 / -
Logistics domain	short	9.4 / 79.4	7.6 / 97.6
	long	13.1 / 74.0	12.6 / 96.0

4.2. Runtime Benchmark

This section evaluates the runtime performance of Paraformer-large in terms of both CER and RTF on an Intel(R) Xeon(R) CPU E5-8269CY @ 2.5GHz with one thread core. We evaluate the performance of Paraformer-large with two runtime backends, as shown in Table 8. To optimize performance, FunASR adopts the automatic mixed precision quantization (AMP) method proposed in [25, 31]. The results demonstrate that AMP quantization improves inference speed by 40% without any significant reduction in recognition accuracy.

Table 8: *Runtime benchmark.*

Quantization	Libtorch		Onnx	
	Float32	Int8	Float32	Int8
RTF	0.1026	0.0597	0.0778	0.0446
CER	1.95	1.95	1.95	1.95

5. Conclusions

This paper presents FunASR, a system designed to bridge the gap between academic research and industrial applications in speech recognition. FunASR provides access to models trained on large-scale industrial corpora, as well as the ability to easily deploy them in real-world applications. We make a wide range of industrial models available, including Paraformer-large model, as well as FSMN-VAD and CT-Transformer models, etc. By making these models openly available, FunASR enables researchers to easily deploy them in real-world scenarios.

6. References

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[2] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "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)*. IEEE, 2016, pp. 4960–4964.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[5] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Proc. Interspeech 2020*, pp. 5036–5040, 2020.

[6] Z. Gao, S. Zhang, M. Lei, and I. McLoughlin, "San-m: Memory equipped self-attention for end-to-end speech recognition," *arXiv preprint arXiv:2006.01713*, 2020.

[7] S. Zhang, Z. Gao, H. Luo, M. Lei, J. Gao, Z. Yan, and L. Xie, "Streaming chunk-aware multihead attention for online end-to-end speech recognition," *arXiv preprint arXiv:2006.01712*, 2020.

[8] Z. Gao, S. Zhang, M. Lei, and I. McLoughlin, "Universal asr: Unifying streaming and non-streaming asr using a single encoder-decoder model," *arXiv preprint arXiv:2010.14099*, 2020.

[9] M. Radfar, R. Barnwal, R. V. Swaminathan, F.-J. Chang, G. P. Strimel, N. Susanj, and A. Mouchtaris, "ConvRNN-T: Convolutional Augmented Recurrent Neural Network Transducers for Streaming Speech Recognition," in *Proc. Interspeech 2022*, 2022, pp. 4431–4435.

[10] I. Sklyar, A. Piunova, and C. Osendorfer, "Separator-Transducer-Segmenter: Streaming Recognition and Segmentation of Multi-party Speech," in *Proc. Interspeech 2022*, 2022, pp. 4451–4455.

[11] C.-T. Do, M. Li, and R. Doddipatla, "Multiple-hypothesis RNN-T Loss for Unsupervised Fine-tuning and Self-training of Neural Transducer," in *Proc. Interspeech 2022*, 2022, pp. 4446–4450.

[12] J. Lee, L. Lee, and S. Watanabe, "Memory-Efficient Training of RNN-Transducer with Sampled Softmax," in *Proc. Interspeech 2022*, 2022, pp. 4441–4445.

[13] K. Zhao, H. Nguyen, A. Jain, N. Susanj, A. Mouchtaris, L. Gupta, and M. Zhao, "Knowledge Distillation via Module Replacing for Automatic Speech Recognition with Recurrent Neural Network Transducer," in *Proc. Interspeech 2022*, 2022, pp. 4436–4440.

[14] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[15] Z. Yao, D. Wu, X. Wang, B. Zhang, F. Yu, C. Yang, Z. Peng, X. Chen, L. Xie, and X. Lei, "Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit," *arXiv preprint arXiv:2102.01547*, 2021.

[16] H. Zhang, T. Yuan, J. Chen, X. Li, R. Zheng, Y. Huang, X. Chen, E. Gong, Z. Chen, X. Hu *et al.*, "Paddlespeech: An easy-to-use all-in-one speech toolkit," *arXiv preprint arXiv:2205.12007*, 2022.

[17] W. Kang, L. Guo, F. Kuang, L. Lin, M. Luo, Z. Yao, X. Yang, P. Żelasko, and D. Povey, "Fast and parallel decoding for transducer," *arXiv preprint arXiv:2211.00484*, 2022.

[18] Z. Gao, S. Zhang, I. McLoughlin, and Z. Yan, "Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition," *arXiv preprint arXiv:2206.08317*, 2022.

[19] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.

[20] J. Du, X. Na, X. Liu, and H. Bu, "Aishell-2: transforming Mandarin asr research into industrial scale," *arXiv preprint arXiv:1808.10583*, 2018.

[21] B. Zhang, H. Lv, P. Guo, Q. Shao, C. Yang, L. Xie, X. Xu, H. Bu, X. Chen, C. Zeng *et al.*, "Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6182–6186.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[23] S. Zhang, M. Lei, Z. Yan, and L. Dai, "Deep-fsmn for large vocabulary continuous speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5869–5873.

[24] Q. Chen, M. Chen, B. Li, and W. Wang, "Controllable time-delay transformer for real-time punctuation prediction and disfluency detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8069–8073.

[25] Z. Gao, Y. Yao, S. Zhang, J. Yang, M. Lei, and I. McLoughlin, "Extremely Low Footprint End-to-End ASR System for Smart Device," in *Proc. Interspeech 2021*, 2021, pp. 4548–4552.

[26] L. Dong and B. Xu, "CIF: Continuous integrate-and-fire for end-to-end speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6079–6083.

[27] X. Shi, Y. Chen, S. Zhang, and Z. Yan, "Achieving timestamp prediction while recognizing with non-autoregressive end-to-end asr model," *arXiv preprint arXiv:2301.12343*, 2023.

[28] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[29] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: end-to-end contextual speech recognition," in *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018, pp. 418–425.

[30] B. Zhang, D. Wu, Z. Peng, X. Song, Z. Yao, H. Lv, L. Xie, C. Yang, F. Pan, and J. Niu, "Wenet 2.0: More productive end-to-end speech recognition toolkit," *arXiv preprint arXiv:2203.15455*, 2022.

[31] K. Zhu, W. Zhao, Z. Zheng, T. Guo, P. Zhao, J. Bai, J. Yang, X. Liu, L. Diao, and W. Lin, "Disc: A dynamic shape compiler for machine learning workloads," in *Proceedings of the 1st Workshop on Machine Learning and Systems*, 2021, pp. 89–95.