INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Universal Speech Transformer

Yingzhu Zhao[1,2*], *Chongjia Ni*[2], *Cheung-Chi Leung*[2], *Shafiq Joty*[1], *Eng Siong Chng*[1], *Bin Ma*[2]

[1]Nanyang Technological University, Singapore
[2]Machine Intelligence Technology, Alibaba Group

{srjoty, aseschng}@ntu.edu.sg
{yingzhu.zhao, ni.chongjia, cc.leung, b.ma}@alibaba-inc.com

Abstract

Transformer model has made great progress in speech recognition. However, compared with models with iterative computation, transformer model has fixed encoder and decoder depth, thus losing the recurrent inductive bias. Besides, finding the optimal number of layers involves trial-and-error attempts. In this paper, the universal speech transformer is proposed, which to the best of our knowledge, is the first work to use universal transformer for speech recognition. It generalizes the speech transformer with dynamic numbers of encoder/decoder layers, which can relieve the burden of tuning depth related hyperparameters. Universal transformer adds the depth and positional embeddings repeatedly for each layer, which dilutes the acoustic information carried by hidden representation, and it also performs a partial update of hidden vectors between layers, which is less efficient especially on the very deep models. For better use of universal transformer, we modify its processing framework by removing the depth embedding and only adding the positional embedding once at transformer encoder frontend. Furthermore, to update the hidden vectors efficiently, especially on the very deep models, we adopt a full update. Experiments on LibriSpeech, Switchboard and AISHELL-1 datasets show that our model outperforms a baseline by 3.88%-13.7%, and surpasses other model with less computation cost.

Index Terms: speech recognition, universal transformer, dynamic depth, recurrent inductive bias

1. Introduction

End-to-end models have been introduced into automatic speech recognition (ASR) successfully over conventional hybrid models. Being a single model which directly maps audio signals to text sequences, end-to-end models learn the mapping holistically without error propagation from multiple pipelined components or using handcrafted features like pronunciation dictionary, greatly reducing system complexity. In recent years, a number of end-to-end models have been proposed for ASR including connectionist temporal classification (CTC) models [1, 2, 3], attention based encoder-decoder models [4, 5, 6], and RNN transducer [7]. CTC models map audio signals to target labels by an encoder only, which is quite straightforward. However, it ignores the interdependence between speech frames, thus missing the contextual information. Attention based encoder-decoder models are the most frequently used ones. The encoder transforms audio signals into high-level representations, from which the decoder generates the text sequences in an auto-regressive manner one token at a time.

More recently, the transformer model has been brought into ASR and is named as *speech transformer* [8]. It is an encoder-decoder architecture as well, except that it uses self-attention network instead of convolutional neural network (CNN) or recurrent neural network (RNN). Self-attention network can learn pairwise relationship between any two elements directly. It does not incur the vanishing gradient problem of RNN or is limited by the kernel size of CNN, thus is able to capture longer range context. Besides, its capability for parallel computation also enables batched operation and fast computation speed. Several further studies use speech transformer model for Mandarin Chinese speech recognition [9, 10] and online speech recognition [11].

Despite the advantages mentioned above, the fixed numbers of encoder and decoder layers in the transformer model limit its computation capability. On the one hand, compared with RNN and long short-term memory (LSTM) networks which have iterative or recursive computation, speech transformer model loses the recurrent inductive bias, which is helpful to tackle tasks of varying complexity. Each input speech time step goes through the same and fixed numbers of encoder and decoder layers to compute the final output, regardless of the fact that different speech time steps differ in phoneme obscurity and noise level, thus may require different computation resources. On the other hand, determining the numbers of encoder and decoder layers requires careful tuning for each dataset to achieve the optimal performance. Speech transformer model was tested for performance in [8] with 5 different depth combinations of encoder and decoder.

There are several studies dealing with the depth of encoder and decoder building blocks. In [6], it builds a deeper encoder by adding residual blocks and using multiple CNN layers before bidirectional LSTM network. The time-depth separable convolution model [12] trains very deep convolutional encoders via a soft attention window pre-training scheme. [13] proposes a very deep transformer architecture with up to 48 encoder and decoder layers to enlarge the computation capability. It also applies stochastic residual layers to improve generalizability and to prevent overfitting. However, these methods still require us to tune depth related hyperparameters, e.g. [13] tested 9 different depth combinations with depth ranging from 4 layers to 48 layers. Different from [13], [14] trains very deep transformer models (up to 40 layers) and then randomly drop layers at training time in order to do efficient layer pruning at inference time. Recently, M. Dehghani et al. [15] proposes the universal transformer model, which achieves the state-of-the-art results on LAMBADA language modeling task. It has the transformer-like architecture and uses a dynamic per-position halting mechanism to choose the required number of layers for each input time step dynamically, which exactly addresses the issues with speech transformer analyzed above.

In this paper, we successfully introduce the universal transformer model to ASR task and we term our model as *universal*

*Yingzhu Zhao is under the Joint PhD Program between Alibaba and Nanyang Technological University.

http://dx.doi.org/10.21437/Interspeech.2020-1716

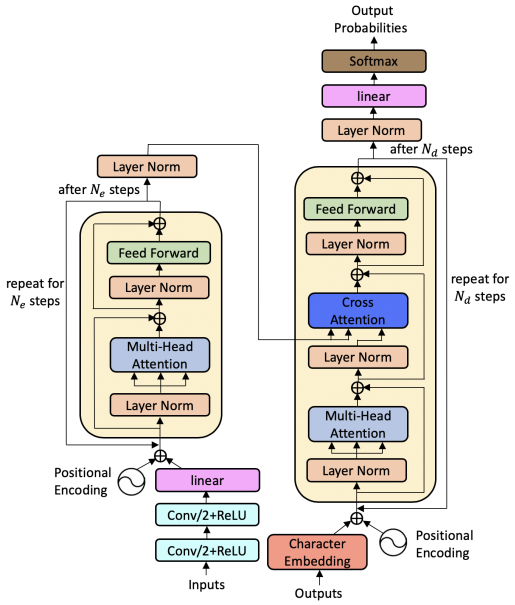

Figure 1: *Universal Speech Transformer model architecture.*

speech transformer. To the best of our knowledge, this is the first work regarding the dynamic encoder and decoder depth in ASR. The recurrent nature of universal transformer best suits the needs of recognizing phonemes with different complexity and noise level, at the same time dynamically learning the encoder and decoder depth, which relieves the burden of tuning depth related hyperparameters. However, universal transformer model has two problems when applied on ASR. First, it adds the depth embedding and positional embedding repeatedly for each layer, which dilutes the acoustic information carried by hidden representation. Second, it performs a partial update of hidden vectors between layers, which is less efficient compared to the full update given the same number of update. To tackle these two problems, we remove the depth embedding and only add the positional embedding once at the transformer encoder frontend, and we replace the partial update of hidden representations between layers with a full update. On the LibriSpeech, Switchboard and AISHELL-1 ASR datasets, our proposed universal speech transformer model outperforms a baseline by 3.88%-13.7%, and achieves better results with much less computation cost compared with the very deep transformer model using 36 encoder layers and 12 decoder layers in [13]. From the experimental results, it can be seen that the number of encoder layers required varies among different input time steps and different datasets, which further substantiates the value of dynamic depth over fixed depth for datasets with varying complexity.

2. Model architecture

2.1. Universal speech transformer

Universal speech transformer is based on the popular speech transformer model, which we refer the reader to [8] for full details. Same as speech transformer, the core module of universal speech transformer is the multi-head attention network. The main change is on the dynamic encoder and decoder depth. Speech transformer model has fixed encoder and decoder depth. Compared with RNN and LSTM networks which have iterative or recursive computation, speech transformer loses the recurrent inductive bias. Universal speech transformer addresses this

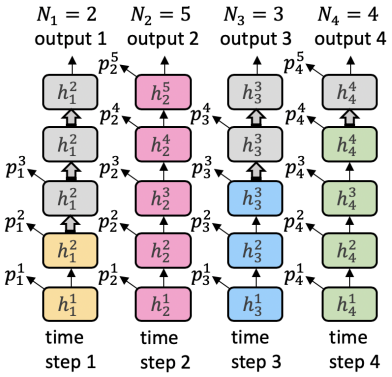

Figure 2: *Example of adaptive computation time technique with 4 input time steps. Maximum number of layers L is 5 here. Illustration is applicable to both encoder and decoder.*

issue with dynamic encoder and decoder depth. The overall framework of universal speech transformer is presented in Figure 1. The adaptive computation time technique [16] is applied to each input time step to calculate the required computation resources, in this case the numbers of encoder and decoder layers, before emitting the final outputs. Each input speech time step requires different level of computation resources due to obscurity of different phonemes and noise variation along the utterance. So does each text input. In particular, for the hidden state $H^j = (h_1^j, ..., h_t^j) \in \mathbb{R}^{t \times d}$ in j^{th} layer, a probability vector at i^{th} time step is calculated by a sigmoid function:

$$p_i^j = k\sigma(Wh_i^j + b) \tag{1}$$

where $W \in \mathbb{R}^{d \times 1}$, $b \in \mathbb{R}^1$, k is a scaling factor, W, b and k are shared across all layers, $i \in [1, t]$, $j \in [1, L]$, L is the maximum depth defined beforehand.

The halting probability is the summation of the probability calculated in Eq. 1, which denotes the probability to emit the final output at i^{th} time step in j^{th} layer:

$$halt_i^j = \sum_{k=1}^{j} p_i^k \tag{2}$$

The number of encoder or decoder layers required at each input time step is decided when the halting probability reaches the threshold, or when the encoder or decoder reaches the maximum depth L:

$$N_i = min(L, max(n' : halt_i^{n'} \leq 1 - \epsilon)) \tag{3}$$

where ϵ is a small constant (0.01 in this paper).

The number of encoder or decoder layers is therefore the maximum number of layers among all input time steps:

$$N = max(N_i) \quad i \in [1, t] \tag{4}$$

The input at each time step i emits the corresponding output at N_i^{th} layer. For the input time steps where the outputs are emitted earlier than the other time steps, i.e. $N_i < N$, the last hidden states are carried forward until all the time steps emit the outputs or when the encoder or decoder reaches the maximum depth. The hidden state at i^{th} time step in j^{th} layer is thus:

$$h_i^j = \begin{cases} h_i^j & 1 \leq j \leq N_i \\ h_i^{N_i} & N_i < j \leq N \end{cases} \tag{5}$$

We illustrate the adaptive computation time technique for calculation of the numbers of encoder and decoder layers in Figure 2. Each hidden state computes a probability to determine whether it halts at current layer, i.e. reach the required number of layers. Hidden state shaded in grey means the current time step halts already, but because there are other time steps which have not reached the required numbers of layers, this time step hidden state is replicated over to the next layer (by thick arrow). Otherwise, the computation between each layer is through the standard transition function (by thin arrow), in this case the transformer encoder or decoder multi-head attention and feedforward network.

2.2. Modifications on universal transformer model

As mentioned earlier, universal transformer model has two problems when applied on ASR. Therefore, we make two modifications to the original universal transformer model here. Firstly, universal transformer model adds both the positional embedding and the depth embedding repeatedly for each encoder and decoder layer as Eq. 6 and Eq. 7.

$$PE^{dep}_{(pos,2i)} = sin(pos/10000^{2i/d}) + sin(dep/10000^{2i/d}) \quad (6)$$

$$PE^{dep}_{(pos,2i+1)} = cos(pos/10000^{2i/d}) + cos(dep/10000^{2i/d}) \quad (7)$$

where pos is the position, dep is the depth, d is the positional and depth embedding dimension, and i is the i^{th} dimension.

For the universal speech transformer, we do not add the depth embedding for each encoder and decoder layer. Only the positional embedding is added once before the repeated building blocks as Eq. 8 and Eq. 9. This is based on the assumption that adding the positional and depth embeddings iteratively for each layer will dilute the acoustic information carried by hidden representation, given that each speech frame under analysis only has a window size of 25ms and does not contain much time-domain information. In contrast, universal transformer encodes the word embedding. A word possibly spans tens of analysis frames in ASR context, so a word embedding contains much richer information, and it will be of less influence to repeatedly add the positional and depth embeddings. Besides, the average depth in our model (around 21) is much higher than the average depth in the universal transformer model (around 8 in LAMBADA language modeling), so adding the depth embedding repeatedly for each layer has more impact in our model. Additionally, adding the depth embedding to each layer provides the depth information to the hidden representation and may be beneficial for calculating the optimal encoder and decoder depth. However, depth information is already implicitly embedded in the hidden representation, which is calculated progressively from layer to layer. Removing the depth embedding does not affect the dynamic encoder and decoder depth computation.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d}) \quad (8)$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d}) \quad (9)$$

Secondly, in each layer of the universal transformer model, hidden representation is updated as below using p^j_i calculated from Eq. 1:

$$H^{j+1} = [p^j_1, ..., p^j_t] * \tilde{H}^{j+1} + (1 - [p^j_1, ..., p^j_t]) * H^j \quad (10)$$

where $\tilde{H}^{j+1}$ is the transformed state after multi-head attention and feedforward network and H^{j+1} is the updated state, i.e. it only performs a partial update of hidden state by a p^j_i probability, with the addition of previous hidden state by a $1 - p^j_i$ probability. The partial update is not very efficient, and calculating hidden state as Eq. 10 requires more times of update and deeper layers to reach the optimal hidden state, which in turn may bring the vanishing gradient problem. To avoid this problem, and given that our model is already very deep, we use the transformed state $\tilde{H}^{j+1}$ directly as the next layer hidden representation without the probability factor to perform an efficient full update. We will test these two modifications in experiments.

Table 1: *Details of datasets used for experiments*

LibriSpeech	
Training set	100h
Test_clean set	5.4h
Test_other set	5.1h
Switchboard	
Training set	300h
SWBD set	2.1h
CallHome set	1.6h
AISHELL-1	
Training set	150h
Test set	5h

Table 2: *WER results of end-to-end speech recognition models on LibriSpeech 100h*

Model	Test_clean	Test_other
End-to-end (E2E) [17]	14.7	40.8
E2E with augmented data [18]	15.1	-
LAS [19]	12.9	35.5
Baseline	12.0	29.7

3. Experiments

3.1. Datasets

We conduct experiments with the universal speech transformer model on three publicly available datasets, including LibriSpeech [20], Switchboard [21] and AISHELL-1 [22]. LibriSpeech consists of 16kHz read English speech from audiobooks. Switchboard is a 300 hour corpus of conversational English telephone speech. AISHELL-1 is a 16kHz Chinese Mandarin speech corpus recorded by 400 speakers from different accent areas in China. The characteristics of the datasets are summarized in Table 1.

3.2. Experimental setup

We use Espnet toolkit [23] for experiments. Input acoustic features are 80-dimensional filterbanks extracted with a window size of 25ms shifted every 10ms, which are mean and variance normalized. Utterances longer than 3000 frames or 400 characters are discarded to keep memory manageable. In the training stage, the input samples are shuffled randomly and trained with batch size 12. We adopt the unigram sub-word algorithm with maximum vocabulary size being 5000. The two CNN layers at the bottom of encoder in Figure 1 have filter size (3,2) and stride 2, each followed by a rectified linear unit (ReLU) activation. For multi-head attention network, the attention dimension is 256, the number of attention heads is 4, the dimension of feedforward network is 2048. For universal speech transformer

Table 3: *Two modifications on universal transformer model on LibriSpeech 100h*

Model	Test_clean	Test_other
Baseline	12.0	29.7
Universal transformer	24.2	43.7
The proposed model	**11.3**	**28.3**
w/o remove depth embedding	19.5	37.8
w/o full hidden state update	15.0	32.1

Table 4: *Comparison with very deep transformer model on all three datasets*

Model	Switchboard (WER)		
	SWBD	CallHome	All
Baseline	8.8	18.0	13.4
36Enc-12Dec [13]	10.4	18.6	-
48Enc-12Dec [13]	10.7	19.4	-
60Enc-12Dec [13]	10.6	19.0	-
Ensemble of above 3 [13]	9.9	17.7	-
The proposed model	**8.3**	**17.3**	**12.8**
Model	**AISHELL-1 (CER)**		
	Dev	**Test**	
Baseline	6.4	7.3	
36Enc-12Dec [13]	6.4	7.3	
The proposed model	**5.8**	**6.3**	
Model	**LibriSpeech 100h (WER)**		
	Test_clean	**Test_other**	
Baseline	12.0	29.7	
36Enc-12Dec [13]	12.8	30.4	
The proposed model	**11.3**	**28.3**	

related hyperparameters, the maximum depth L in encoder is 24, and the maximum depth L in decoder is 16, the scaling factor k is set as 0.25. During experiments, the dynamic depth for encoder and decoder obtained tends to be small, so we set a minimum depth for encoder and decoder, i.e. the model only performs dynamic depth computation after the minimum depth is reached. Based on the encoder and decoder depth combinations tested by [13], we set the minimum depth for encoder to 10, and 6 for decoder. The initial value of the learning rate is 5.0, the encoder and decoder dropout rate is 0.1. We have a strong baseline as shown in Table 2, which comes from the best results well designed in Espnet toolkit.

3.3. Results

We first test the two modifications on the universal transformer model analyzed in Section 2.2 using LibriSpeech 100h dataset. From Table 3, using universal transformer model directly for speech recognition deteriorates performance a lot compared to the baseline. The universal speech transformer model using the efficient full hidden state update and removing the depth embedding for each layer achieves the best performance. We believe that using the full hidden state update facilitates the encoder or decoder to obtain the optimal hidden representation. Removing the depth embedding is most efficient, and it confirms our assumption that adding depth embedding repeatedly for each layer dilutes the acoustic information in hidden representation.

Next, we compare our proposed method with [13], which

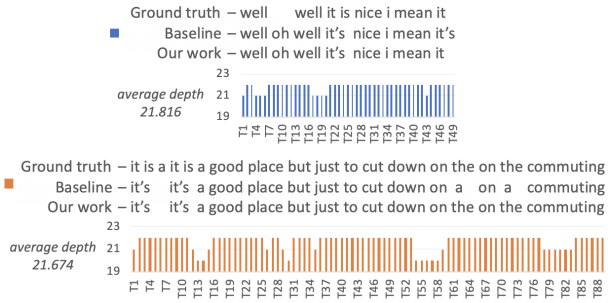

Figure 3: *Number of encoder layers required N_i in Eq. 3 for each time step i and average encoder depth of two randomly sampled speech utterances from Switchboard test dataset.*

Table 5: *Number of utterances and average encoder depth across all speech time steps for three datasets*

Dataset	No. of utterances	Depth
LibriSpeech		
Test_clean	2620	21.118
Test_other	2939	21.268
Switchboard		
All	4458	21.738
AISHELL-1		
Test	5741	21.428

did 9 test runs and set a good benchmark for various encoder and decoder depth combinations on transformer model. The results of three datasets are summarized in Table 4. [13] tested 9 encoder and decoder depth combinations ranging from 4 layers to 48 layers, and the best combination is using 36 encoder layers and 12 decoder layers. They further did an ensemble of three models (row 2, 3, 4 of Switchboard experiments) to achieve their optimal performance (row 5 of Switchboard experiments). Our model surpasses the baseline by 3.88%-13.7%, and outperforms [13] on all three datasets. We randomly sample two utterances from Switchboard dataset and compare the recognition result with the baseline in Figure 3. It can be seen that for both cases, wrong words near the end of the utterances predicted by the baseline are corrected by our model with relatively higher number of layers in those time steps. In addition, we calculate the average encoder depth across all speech time steps for three datasets and list them in Table 5. The average numbers of encoder layers are all around 21 in three datasets. Compared with [13] which deploys 36 encoder layers, our model dynamically determines the encoder and decoder depth for each input time step with significantly less training cost overall, which further demonstrates the value and potential of universal speech transformer model.

4. Conclusions

In this paper, we introduce the universal speech transformer, which generalizes speech transformer with dynamic encoder and decoder depth for each input time step. Our model is capable of tackling tasks of varying complexity by bringing the recurrent inductive bias to speech transformer model, as well as relieving the burden of tuning depth related hyperparameters. It outperforms the baseline by 3.88%-13.7%, and achieves better performance than 36 layer encoder model with much less computation cost. In the future, we are interested in exploring computation resource differences between different languages.

5. References

[1] 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 (ICASSP)*. IEEE, 2013, pp. 6645–6649.

[2] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning (ICML)*, 2014, pp. 1764–1772.

[3] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *2015 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015, pp. 167–174.

[4] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[5] W. Chan, N. Jaitly, Q. V. 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.

[6] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4845–4849.

[7] A. Graves, "Sequence transduction with recurrent neural networks," in *International Conference of Machine Learning (ICML)*, 2012, pp. 235–242.

[8] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[9] Y. Zhao, J. Li, X. Wang, and Y. Li, "The speechtransformer for large-scale mandarin chinese speech recognition," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[10] S. Zhou, L. Dong, S. Xu, and B. Xu, "A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on mandarin chinese," *arXiv preprint arXiv:1805.06239*, 2018.

[11] L. Dong, F. Wang, and B. Xu, "Self-attention aligner: A latency-control end-to-end model for ASR using self-attention network and chunk-hopping," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[12] A. Hannun, A. Lee, Q. Xu, and R. Collobert, "Sequence-to-sequence speech recognition with time-depth separable convolutions," in *INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, 2019.

[13] N.-Q. Pham, T.-S. Nguyen, J. Niehues, M. Müller, S. Stüker, and A. Waibel, "Very deep self-attention networks for end-to-end speech recognition," in *INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, 2019.

[14] A. Fan, E. Grave, and A. Joulin, "Reducing transformer depth on demand with structured dropout," in *2020 International Conference on Learning Representations (ICLR)*, 2020.

[15] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Łukasz Kaiser, "Universal transformers," in *2019 International Conference on Learning Representations (ICLR)*, 2019.

[16] A. Graves, "Adaptive computation time for recurrent neural networks," *arXiv preprint arXiv:1603.08983*, 2016.

[17] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "RWTH asr systems for librispeech: Hybrid vs attention," in *INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, 2019, pp. 231–235.

[18] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[19] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "On the choice of modeling unit for sequence-to-sequence speech recognition," in *INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, 2019.

[20] 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.

[21] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: telephone speech corpus for research and development," in *1992 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1992.

[22] 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.

[23] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *INTERSPEECH 2018 – 19th Annual Conference of the International Speech Communication Association*, 2018, pp. 2207–2211.