TRANSFORMER-BASED ONLINE CTC/ATTENTION END-TO-END SPEECH RECOGNITION ARCHITECTURE

Haoran Miao[1,2], *Gaofeng Cheng*[1], *Changfeng Gao*[1,2], *Pengyuan Zhang*[1,2], *Yonghong Yan*[1,2,3]

[1]Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, China
[2]University of Chinese Academy of Sciences, China
[3]Xinjiang Laboratory of Minority Speech and Language Information Processing, Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, China

ABSTRACT

Recently, Transformer has gained success in automatic speech recognition (ASR) field. However, it is challenging to deploy a Transformer-based end-to-end (E2E) model for online speech recognition. In this paper, we propose the Transformer-based online CTC/attention E2E ASR architecture, which contains the chunk self-attention encoder (chunk-SAE) and the monotonic truncated attention (MTA) based self-attention decoder (SAD). Firstly, the chunk-SAE splits the speech into isolated chunks. To reduce the computational cost and improve the performance, we propose the state reuse chunk-SAE. Sencondly, the MTA based SAD truncates the speech features monotonically and performs attention on the truncated features. To support the online recognition, we integrate the state reuse chunk-SAE and the MTA based SAD into online CTC/attention architecture. We evaluate the proposed online models on the HKUST Mandarin ASR benchmark and achieve a 23.66% character error rate (CER) with a 320 ms latency. Our online model yields as little as 0.19% absolute CER degradation compared with the offline baseline, and achieves significant improvement over our prior work on Long Short-Term Memory (LSTM) based online E2E models.

Index Terms— Transformer, end-to-end speech recognition, online speech recognition, CTC/attention speech recognition

1. INTRODUCTION

In recent years, the end-to-end (E2E) automatic speech recognition (ASR) has gained popularity in ASR community [1, 2, 3, 4, 5, 6]. E2E ASR models simplify the hybrid DNN/HMM ASR models by replacing the acoustic, pronunciation and language models with one single deep neural network, and thus transcribe speech to text directly. To date, E2E ASR models have achieved significant improvement in ASR field [4, 5, 6]. The hybrid Connectionist Temporal Classification (CTC) / attention E2E ASR architecture [6] has attracted lots of attention because it combines the advantages of CTC models and attention models. During training, the CTC objective is attached to the attention-based encoder-decoder model as an auxiliary task. During decoding, the joint CTC/attention decoding approach is adopted in the beam search [7]. However, it is difficult to deploy the online CTC/attention E2E ASR architecture because of global attention mechanisms [8] and CTC prefix scores [6, 9], which depend on the entire input speech. Our prior work [10, 11] has streamed this architecture from both the model structure and decoding algorithm aspects. On the model structure aspect, we proposed the stable monotonic chunk-wise attention (sMoChA) [10] and monotonic truncated attention (MTA) [11] to stream attention mechanisms, and applied the latency-controlled bidirectional long short-term memory (LC-BLSTM) as the low-latency encoder. On the decoding aspect, we proposed the online joint decoding approach, which includes truncated CTC (T-CTC) prefix scores and dynamic waiting joint decoding (DWJD) algorithm [10].

Recently, Transformer [12] has gained success in ASR field [13, 14, 15]. Transformer-based models are parallelizable and competitive to recurrent neural networks [16]. However, the vanilla Transformer is inapplicable to online tasks for two reasons: First, the self-attention encoder (SAE) computes the attention weights on the whole input frames; Second, the self-attention decoder (SAD) computes the attention weights on the whole outputs of SAE.

In this paper, we stream the Transformer and integrate it into the CTC/attention E2E ASR architecture. On the SAE aspect, we propose the chunk-SAE which splits the input speech into isolated chunks of fixed length. Inspired by Transformer-XL [17], we further propose the state reuse chunk-SAE which reuses the stored states of the previous chunks to reduce the computational cost. On the SAD aspect, we propose the MTA based SAD, which performs attention on the truncated historical outputs of SAE. Finally, we propose the Transformer-based online CTC/attention E2E ASR architecture via the online joint decoding approach [10]. Our experiments shows that the proposed online model with a 320 ms latency achieves 23.66% character error rate (CER) on HKUST, with only 0.19% absolute CER degradation compared with the offline baseline.

The rest of this paper is organized as follows. In Section 2, we describe the online CTC/attention E2E architecture proposed in our prior work [10, 11]. In Section 3, we introduce the Transformer architecture. In Section 4, we describe the online Transformer-based CTC/attention architecture. The experiments and conclusions are presented in Sections 5 and 6, respectively.

2. ONLINE CTC/ATTENTION E2E ARCHITECTURE

In our prior work [10], we proposed an online hybrid CTC/attention E2E ASR architecture, which consists of the LC-BLSTM encoder, sMoChA and LSTM decoder. During training, we introduce the CTC objective as an auxiliary task, and the loss function is defined

This work is partially supported by the National Key Research and Development Program (Nos. 2018YFC0823402, 2018YFC0823401, 2018YFC0823405, 2018YFC0823400), the National Natural Science Foundation of China (Nos. 11590774, 11590772, 11590770),the Key Science and Technology Project of the Xinjiang Uygur Autonomous Region (No.2016A03007-1).

by:

$$\mathcal{L} = \alpha\mathcal{L}_{\rm dec} + (1-\alpha)\mathcal{L}_{\rm ctc}, \quad (1)$$

where α is a hyperparameter, $\mathcal{L}_{\rm dec}$ and $\mathcal{L}_{\rm ctc}$ are loss functions from the decoder and CTC. During decoding, we adopt the online joint decoding approach, which is defined by:

$$\hat{Y} = \arg\max_{Y\in\mathcal{Y}^*}\{\lambda\log P_{\rm dec}(Y|X) + (1-\lambda)\log P_{\text{t-ctc}}(Y|X) + \gamma\log P_{\rm lm}(Y)\}, \quad (2)$$

where $P_{\rm dec}(Y|X)$ and $P_{\text{t-ctc}}(Y|X)$ are the probabilities of the hypothesis Y conditioned on input frames X from the decoder and T-CTC [10], and $P_{\rm lm}(Y)$ is the language model probability. The hyperparameters λ and γ are tunable. For online decoding, we proposed DWJD algorithm [10] to 1) coordinate the forward propagation in the encoder and the beam search in the decoder; 2) address the unsynchronized predictions of the sMoChA-based decoder and CTC outputs.

MTA [11], which performs attention on top of the truncated historical encoder outputs, outperforms the sMoChA by exploiting longer history. Formally, we denote $\mathbf{q}_i$ and $\mathbf{h}_j$ as the i-th decoder state and the j-th encoder output, respectively. Similar to monotonic chunk-wise attention [18], MTA defines the probability $p_{i,j}$ of truncating encoder outputs at $\mathbf{h}_j$ as:

$$p_{i,j} = \text{Sigmoid}(g\frac{\mathbf{v}^\top}{||\mathbf{v}||}\tanh(\mathbf{W}_1\mathbf{q}_{i-1} + \mathbf{W}_2\mathbf{h}_j + \mathbf{b}) + r), \quad (3)$$

where the matrices $\mathbf{W}_1$, $\mathbf{W}_2$, vectors $\mathbf{b}$, $\mathbf{v}$ and scalars g, r are trainable parameters. Then, the attention weight $a_{i,j}$ is computed by:

$$a_{i,j} = p_{i,j}\prod_{k=1}^{j-1}(1-p_{i,k}), \quad (4)$$

where $a_{i,j}$ indicates the probability of truncating encoder outputs at $\mathbf{h}_j$ and skipping the encoder outputs before $\mathbf{h}_j$. During decoding, MTA determines a truncation end-point t_i for the i-th decoder step by:

$$z_{i,j} = \mathbb{I}(p_{i,j} > 0.5 \wedge j \geq t_{i-1}), \quad (5)$$

where $z_{i,j}$ denotes the indicator of truncating or do not truncating encoder outputs at $\mathbf{h}_j$, and $\mathbb{I}$ represents an indicator function. By the condition $j \geq t_{i-1}$ in Eq. 5, MTA enforces the end-point to move in a left-to-right mode. Once $z_{i,j} = 1$ for some j, MTA sets t_i to j. Finally, MTA performs attention on the truncated encoder outputs:

$$\mathbf{r}_i = \sum_{j=1}^{t_i} a_{i,j}\mathbf{h}_j, \quad (6)$$

where $\mathbf{r}_i$ is the letter-wise hidden vector for the i-th decoder step.

During training, MTA performs attention on the whole encoder outputs:

$$\mathbf{r}_i = \sum_{j=1}^{T} a_{i,j}\mathbf{h}_j, \quad (7)$$

where T denotes the number of encoder outputs.

3. TRANSFORMER ARCHITECTURE

Transformer [12] follows the encoder-decoder architecture using stacked self-attention and position-wise feed-forward layers for both the encoder and decoder. We briefly introduce the Transformer architecture in this section.

3.1. Multi-head attention

Transformer adopts the scaled dot-product attention to map a query and a set of key-value pairs to an output as:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_m}})\mathbf{V}, \quad (8)$$

where the matrices $\mathbf{Q} \in \mathbb{R}^{n\times d_m}$, $\mathbf{K} \in \mathbb{R}^{m\times d_m}$ and $\mathbf{V} \in \mathbb{R}^{m\times d_m}$ denote queries, keys and values, n and m denote the number of queries and keys (or values), and d_m denotes representation dimension.

Instead of performing a single attention function, Transformer uses multi-head attention that jointly learns diverse relationships between queries and keys from different representation sub-spaces as follows:

$$\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(\text{head}_1,\cdots,\text{head}_H)\mathbf{W}^O, \quad (9)$$

$$\text{head}_h = \text{Attention}(\mathbf{Q}\mathbf{W}_h^Q, \mathbf{K}\mathbf{W}_h^K, \mathbf{V}\mathbf{W}_h^V), \quad (10)$$

where H denotes the head number and $d_k = d_m/H$. The matrices $\mathbf{W}^O \in \mathbb{R}^{d_m\times d_m}$ and $\mathbf{W}_h^{Q,K,V} \in \mathbb{R}^{d_m\times d_k}$ are trainable parameters.

Because Transformer lacks of modeling the sequence order, the work in [12] suggested to use sine and cosine functions of different frequencies to perform the positional encoding.

3.2. Self-attention encoder (SAE)

The SAE consists of a stack of identical layers, each of which has two sub-layers, i.e. one self-attention layer and one position-wise feed-forward layer. The inputs of the SAE are acoustic frames in ASR tasks. The self-attention layer employs multi-head attention, in which the queries, keys and values are inputs of the previous layer. Besides, the SAE uses residual connections [19] and layer normalization [20] after each sub-layer.

3.3. Self-attention decoder (SAD)

The SAD also consists of a stack of identical layers, each of which has three sub-layers, i.e. one self-attention layer, one encoder-decoder attention layer and one position-wise feed-forward layer. The inputs of the SAD are embeddings of right-shifted output labels. To prevent the access to the future output labels in the self-attention, the subsequent positions are masked. In the encoder-decoder attention, the queries are current layer inputs while the keys and values are SAE outputs. Besides, the SAD also uses residual connections and layer normalization after each sub-layer.

4. TRANSFORMER-BASED ONLINE CTC/ATTENTION E2E ARCHITECTURE

In this section, we propose the Transformer-based online E2E model, which consists of the chunk-SAE with or without reusing stored states and MTA based SAD. The Transformer-based online CTC/attention E2E architecture is shown in Fig. 1.

4.1. Chunk-SAE

To stream the SAE, we first propose the chunk-SAE, which splits a speech into non-overlapping isolated chunks of N_c central length. To acquire the contextual information, we splice N_l left frames before each chunk as historical context and N_r right frames after it as

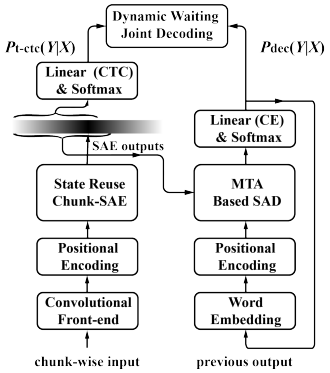

Fig. 1. Transformer-based online CTC/attention E2E architecture.

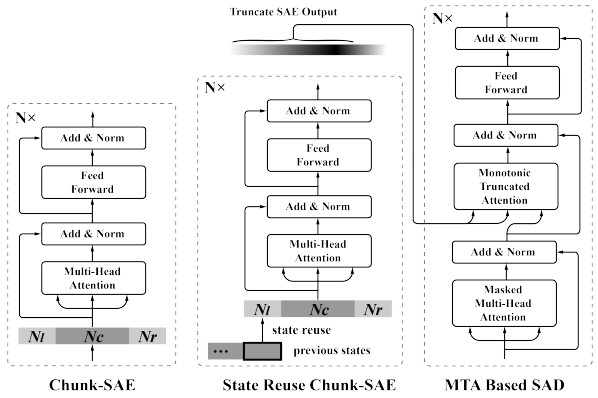

Fig. 2. Illustrations of the chunk-SAE, state reuse chunk-SAE and MTA based SAD.

future context. The spliced frames only act as contexts and give no output. With the predefined parameters N_c, N_l and N_r, the receptive field of each chunk-SAE output is restricted to $N_l + N_c + N_r$ and the latency of the chunk-SAE is limited to N_r.

4.2. State reuse chunk-SAE

In the chunk-SAE, the historical context is re-computed for each chunk. To reduce the computational cost, we store the computed hidden states in central context. Then, when computing the new chunk, we reuse stored hidden states from the previous chunks at the same positions as historical context, which is inspired by Transformer-XL [17]. Fig. 2 illustrates the difference between the chunk-SAE with or without reusing hidden states. Formally, $\mathbf{s}_\tau^l \in \mathbb{R}^{N_l \times d_m}$ and $\mathbf{h}_\tau^l \in \mathbb{R}^{(N_c+N_r)\times d_m}$ denote the stored and newly-computed hidden states for the τ-th chunk in the l-th layer, respectively. Then, the queries, keys and values for the τ-th chunk in the l-th self-attention layer are defined as follows:

$$\mathbf{Q}_\tau^l, \mathbf{K}_\tau^l, \mathbf{V}_\tau^l = \mathbf{h}_\tau^{l-1}, \widetilde{\mathbf{h}}_\tau^{l-1}, \widetilde{\mathbf{h}}_\tau^{l-1}, \quad (11)$$

$$\text{where} \quad \widetilde{\mathbf{h}}_\tau^{l-1} = \text{Concat}(\text{SG}(\mathbf{s}_\tau^{l-1}),\ \mathbf{h}_\tau^{l-1}). \quad (12)$$

In Eq. 12, the function $\text{SG}(\cdot)$ stands for stop-gradient. Therefore, the complexity of the state reuse chunk-SAE is reduced by a factor of $N_l/(N_l + N_c + N_r)$.

Moreover, the state reuse chunk-SAE captures long-term dependency beyond the chunks. Suppose the state reuse chunk-SAE consists of L layers, the receptive field on the left side extends to as far as $L \cdot N_l$ frames, which is much broader than that of chunk-SAE.

4.3. MTA based SAD

To stream the SAD, we propose the MTA based SAD to truncate the receptive field in a monotonic left-to-right way and perform attention on the truncated outputs of SAE. Specifically, we substitute MTA for the encoder-decoder attention in each SAD layer, as shown in Fig. 2. Suppose the representation dimension is d_m, MTA performs in parallel during training as follows:

$$\text{MTA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = (\mathbf{P} \odot \text{cumprod}(\mathbf{1} - \mathbf{P}))\mathbf{V}\mathbf{W}_v, \quad (13)$$

$$\mathbf{P} = \text{sigmoid}(\frac{\mathbf{Q}\mathbf{W}_q {\mathbf{W}_k}^\top \mathbf{K}^\top}{\sqrt{d_m}} + r + \varepsilon), \quad (14)$$

where the matrices $\mathbf{W}_\cdot \in \mathbb{R}^{d_m \times d_m}$ and scalar bias r are trainable parameters, and ϵ denotes the noise. We define $\mathbf{P} = \{p_{i,j}\}$ as the truncation probability matrix, where $p_{i,j}$ indicates the probability of truncating the j-th SAE output in order to predict the i-th output label. In Eq. 13, the cumulative product function $\text{cumprod}(\mathbf{x}) = [1, x_1, x_1x_2, \cdots, \prod_{k=1}^{|x|-1} x_k]$ and $\text{cumprod}(\cdot)$ applies to the rows of $\mathbf{P}$. The notation $\odot$ indicates the element-wise product.

MTA learns the appropriate offset for the pre-sigmoid activations in Eq. 14 via the trainable scalar r. To prevent $\text{cumprod}(\mathbf{1} - \mathbf{P})$ from vanishing to zeros, we initialize r to a negative value, e.g. $r = -4$ in our experiments. To encourage the discreteness of the truncation probabilities, we simply add zero-mean, unit-variance Gaussian noise ε to the pre-sigmoid activations only during training.

During decoding, we have to compute the elements in $\mathbf{P}^l = \{p_{i,j}^l\}$ row by row, where $\mathbf{P}^l$ is the truncation probability matrix in the l-th layer. we define t_i^l as the truncation end-point belonging to the l-th layer when predicting the i-th output label. Then, the end-point is determined by:

$$z_{i,j}^l = \mathbb{I}(p_{i,j}^l > 0.5 \wedge j \geq t_{i-1}^l), \quad (15)$$

where $z_{i,j}^l$ denotes the indicator of truncating or do not truncating j-th SAE output in l-th layer and $\mathbb{I}$ represents an indicator function. Once $z_{i,j}^l = 1$ for some j, we set t_i^l to j, which means that the receptive field of the l-th layer is restricted to t_i^l SAE outputs. Suppose the MTA based SAD consists of L layers, there will be L end-points at each decoding step. The number of truncated SAE outputs in each layer will not affect other layers. Therefore, we define the the maximum of L end-points as the receptive field of the MTA based SAD.

5. EXPERIMENTS

5.1. Corpus

We evaluated our models using HKUST Mandarin Chinese conversational telephone [21]. The HKUST consists of about 200 hours *train* set for training and about 5 hours test set. We extracted 4000 utterances from the *train* set as our development set. To improve the recognition accuracy, we applied the speed perturbation on the rest *train* set by factors 0.9 and 1.1.

5.2. Model descriptions

We built all the online models using ESPnet toolkit [22]. For the input, we used 83-dimensional features, including 80-dimensional fil-

Table 1. The character error rates (CER) of different Transformer-based ASR models on HKUST.

Encoder	Decoder	State Reuse	Encoder Speed Ratio	Dev	Test
SAE	SAD	–	2.8	24.12	23.47
Chunk-	MTA-	×	1.0	24.83	23.74
SAE	SAD	√	1.5	24.45	23.65

ter banks, pitch, delta-pitch and Normalized Cross-Correlation Functions. The features were computed with a 25 ms window and shifted every 10 ms. For the output, we adopted a 3655-sized vocabulary set, including 3623 Chinese Mandarin characters, 26 English characters, as well as 6 non-language symbols denoting laughter, noise, vocalized noise, blank, unknown-character and sos/eos.

We used 2-layer convolutional neural networks (CNN) as the front-end. Each CNN layer had 256 filters, each of which has 3×3 kernel size with 2×2 stride, and thus the time reduction of the front-end was $1/4$. The SAE and SAD had 12 and 6 layers, respectively. All sub-layers, as well as embedding layers, produced outputs of dimension 256. In the multi-head attention networks, the head number was 4. In the position-wise feed-forward networks, the inner dimension was 2048. Besides, we trained a 2-layer 1024-dimensional LSTM network on HKUST transcriptions as the external language model and adopted the above 3655-sized vocabulary set.

During training, we used the CTC/attention joint training ($\alpha = 0.7$) and the Adam optimizer with Noam learning rate schedule (25000 warm steps)[12], and trained for 30 epochs. To prevent overfitting, we used dropout [23] (dropout rate $= 0.1$) in each sub-layer, uniform label smoothing [24] (penalty $= 0.1$) in the output layer and the model averaging approach that averages the parameters of models at the last 10 epochs. During decoding, we adopted online joint decoding approach, combining T-CTC prefix scores ($\lambda = 0.5$) and language model scores ($\gamma = 0.3$) to prune the hypotheses, and the beam size was 10.

5.3. Chunk-SAE with or without reusing states

In Table 1, we compared the speed and performance of the chunk-SAE with or without reusing states. The context configuration remained the same for online models during the comparison, i.e. $N_l = N_c = N_r = 64$. Firstly, we measured the speed of various encoders during decoding using a sever with Intel(R) Xeon(R) Silver 4114 CPU, 2.20GHz. For the clear comparison, we set the speed of chunk-SAE to 1.0 and give the speed ratio of other encoders. In lines 1 and 2 of Table 1, the chunk-SAE was slower than the SAE due to the redundant computation of the historical and future context. In lines 2 and 3 of Table 1, we observed that the state reuse chunk-SAE was 1.5x faster than the chunk-SAE, which is consistent with the theoretical analysis in Section 4.2. In addition to the faster speed, the state reuse chunk-SAE outperformed the chunk-SAE by 1.53% and 0.38% relative CERs reduction on HKUST development and test set, respectively. Because of the faster speed and better performance, we employed the state reuse chunk-SAE in our subsequent experiments.

5.4. Context investigation

in Table 2, we investigated our online model performance varying the historical, central and future context lengths. Firstly, comparing lines 2-4 in Table 2, we can see that the future context brought more improvement than the historical context, which indicates that the future context is more crucial to the performance of our online models. Secondly, comparing lines 5-7 in Table 2, we found that it was effective to increase the length of the historical context when we intended to reduce the latency of the state reuse chunk-SAE and maintain the recognition accuracy at the same time. Thirdly, comparing lines 7 and 8 in Table 2, we found that the CER reduced when we increased the length of central context.

Table 2. The CERs of online Transformer-based ASR models with different context configurations on HKUST.

No.	Model	N_l	N_c	N_r	Dev	Test
1	SAE+SAD	–	–	–	24.12	23.47
2		0	64	0	30.02	28.53
3	State Reuse	64	64	0	29.97	28.41
4	Chunk-SAE	0	64	64	24.94	24.10
5	+	64	64	64	**24.45**	**23.65**
6	MTA-SAD	64	64	32	24.67	24.05
7		96	64	32	24.50	23.66
8		128	32	32	25.04	24.21

Table 3. Comparison with published ASR models on HKUST.

Model	Size	Test
TDNN-hybrid, lattice-free MMI [25]	19M	23.69
Offline Self-attention Aligner [26]	38M*	24.12
Online Self-attention Aligner [26]	24M*	26.52
Offline BLSTM CTC/att model [6]	112M	27.43
Online LC-BLSTM CTC/att model [11]	112M	27.84
Online Transformer-based CTC/att model	31M	**23.66**

* Estimated model parameter size according to model configurations.

Finally, our best online model achieved a 23.65% CER, with a 640 ms latency and a 0.18% absolute CER degradation compared with the offline baseline in line 1 of Table 2. In Table 3, we also compared our Transformer-based online CTC/attention model with other published ASR models. For a fair comparison, the latency of the online E2E models listed in Table 3 is 320 ms. These models were trained on HKUST with speed perturb except online Self-attention Aligner model.

6. CONCLUSION

In this paper, we propose the Transformer-based online E2E ASR model, which consists of the state reuse chunk-SAE and MTA based SAD, and integrate the proposed Transformer-based online E2E ASR model into the CTC/attention ASR architecture. Compared with the simple chunk-SAE, the state reuse chunk-SAE performs better and requires less computational cost, because it has broader historical context via storing the states in previous chunks. Compared with the SAD, the MTA based SAD truncates the SAE outputs in a monotonic left-to-right way and performs attention on the truncated SAE outputs, making it applicable to online recognition. We evaluate the proposed Transformer-based online CTC/attention E2E models on HKUST and achieves a 23.66% CER with a 320 ms latency, which outperforms our prior LSTM-based online E2E models. In future, we plan to adopt teacher-student learning approach to further reduce the model latency.

7. 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*, New York, NY, USA, 2006, ICML '06, pp. 369–376, ACM.

[2] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, Cambridge, MA, USA, 2015, NIPS'15, pp. 577–585, MIT Press.

[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)*, March 2016, pp. 4960–4964.

[4] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.

[5] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 4774–4778.

[6] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, Dec 2017.

[7] T. Hori, S. Watanabe, and J. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, July 2017, pp. 518–529, Association for Computational Linguistics.

[8] D., K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[9] K. Kawakami, *Supervised sequence labelling with recurrent neural networks*, Ph.D. thesis, Ph. D. thesis, Technical University of Munich, 2008.

[10] H. Miao, G. Cheng, P. Zhang, L. Ta, and Y. Yan, "Online Hybrid CTC/Attention Architecture for End-to-End Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2623–2627.

[11] H. Miao and G. Cheng, "Streaming attention," https://github.com/HaoranMiao/streaming-attention, 2020, GitHub repository.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, USA, 2017, NIPS'17, pp. 6000–6010, Curran Associates Inc.

[13] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A norecurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[14] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration," in *Proc. Interspeech 2019*, 2019, pp. 1408–1412.

[15] N. Pham, T. Nguyen, J., M. Mller, and A. Waibel, "Very Deep Self-Attention Networks for End-to-End Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 66–70.

[16] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. Yalta, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs rnn in speech applications," 09 2019.

[17] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *CoRR*, vol. abs/1901.02860, 2019.

[18] C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[20] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[21] Y. Liu, P Fung, Y. Yang, C Cieri, S. Huang, and D. Graff, "Hkust/mts: A very large scale mandarin telephone speech corpus," in *International Conference on Chinese Spoken Language Processing*, 2006.

[22] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, and N. Chen, "Espnet: End-to-end speech processing toolkit," 2018.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,*, 2016.

[25] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *INTERSPEECH*, 2016.

[26] 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 *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5656–5660.