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

Cross Attention with Monotonic Alignment for 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, a state-of-the-art neural network architecture, has been used successfully for different sequence-to-sequence transformation tasks. This model architecture disperses the attention distribution over entire input to learn long-term dependencies, which is important for some sequence-to-sequence tasks, such as neural machine translation and text summarization. However, automatic speech recognition (ASR) has a characteristic to have monotonic alignment between text output and speech input. Techniques like Connectionist Temporal Classification (CTC), RNN Transducer (RNN-T) and Recurrent Neural Aligner (RNA) build on top of this monotonic alignment and use local encoded speech representations for corresponding token prediction. In this paper, we present an effective cross attention biasing technique in transformer that takes monotonic alignment between text output and speech input into consideration by making use of cross attention weights. Specifically, a Gaussian mask is applied on cross attention weights to limit the input speech context range locally given alignment information. We further introduce a regularizer for alignment regularization. Experiments on LibriSpeech dataset find that our proposed model can obtain improved output-input alignment for ASR, and yields 14.5%-25.0% relative word error rate (WER) reductions.

Index Terms: speech recognition, end-to-end, transformer, alignment, cross attention

1. Introduction

Automatic speech recognition (ASR) has made great progress in recent years. From alignment-free CTC models [1, 2, 3], to encoder-decoder attentional models [4, 5, 6], to jointly trained CTC and attention-based models [7, 8, 9], end-to-end models have demonstrated great potential over traditional GMM-HMM and hybrid DNN-HMM models [10, 11]. Recently, the Transformer model [12] has been successfully introduced for ASR. [13] first used the Transformer in speech recognition and named the model as *speech-transformer*. [14] further designed a downsampling method used in the Transformer for speech recognition task. As an end-to-end model architecture, the Transformer not only combines acoustic model, pronunciation dictionary and language model in a unified neural framework, it is also well known for its fast computation speed [13] and ability to learn long range relationships [15].

Transformer model relies on self-attention and cross attention in the encoder and decoder to capture direct pairwise relationships in the respective contexts. Its cross attention from decoder hidden states to encoder hidden states is the same as the cross attention in the long short-term memory (LSTM) based encoder-decoder model, i.e. to attend to the entire input utterances and obtain corresponding attention weights for decoding. This model architecture can disperse the attention distribution over the entire input, which is important for some sequence-to-sequence tasks, such as neural machine translation (NMT) and text summarization. However, when it comes to ASR, the same architecture may not work well, as monotonic alignment between text output and speech input is a characteristic of ASR, and has been studied using various techniques. CTC [16] and its extensions (RNN-T [1], RNA [17]) used monotonic alignment position to locate the local encoder representations for current token prediction. [18] proposed using the CTC output as alignment reference by getting both the first position of consecutive outputs and any non-consecutive output position before the blank label. However, their alignment depends heavily on CTC output accuracy, and the result is not comparable with state-of-the-art end-to-end speech recognition results. The recently proposed Continuous Integrate-and-Fire model [19] followed the integrate-and-fire neural model to integrate along input speech frames and triggered an output once an alignment boundary is found. Their soft and monotonic alignment is affected by background noise greatly, since noise term will influence the integration and alignment boundary location, affecting the final accuracy.

In order to achieve better alignments between output and input for speech recognition under the sequence-to-sequence framework, in this paper we propose a straightforward yet effective cross attention biasing technique for the Transformer model that takes output-input alignments into consideration, without referencing CTC or adding additional parameters on encoder hidden states. We take advantage of cross attention weights as a reference of output-input alignment to be used in current cross attention computation. In particular, we apply a Gaussian mask on attention weights centered at the alignment position. Additionally, we introduce a regularizer which regularizes alignment between output and input to encourage monotonicity. Since lower layers of the Transformer capture more acoustic and local information [20], we apply our cross attention biasing on lower layers of the Transformer model, and leave the cross attention at higher layers to attend to entire speech input to capture global information. Our results on LibriSpeech 100h dataset show that our proposed model yields 14.5%-25.0% relative word error rate (WER) reductions.

2. Model architecture

In this section, we will introduce our proposed model by first reviewing the speech transformer model, and then detailing our proposed approach.

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

http://dx.doi.org/10.21437/Interspeech.2020-1198

2.1. Speech transformer

Transformer was proposed by [12] as an encoder-decoder sequence transduction model. Here we summarize a few key components of the transformer model. For full details, please refer to [12]. Transformer encoder has N_e repeated building blocks and transformer decoder has N_d repeated building blocks, as shown in Figure 1. It replaces the commonly used recurrent layers with self-attention layers. Self-attention network is used in both encoder and decoder, to learn the input representation by scaled dot-product attention. The multiple self-attention outputs are concatenated together to learn different subspaces concurrently, which is called multi-head attention:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \quad (2)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

where h is the number of attention heads, $W_i^Q \in \mathbb{R}^{d_{model} \times d_q}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ are respective weight matrices, and $d_k = d_q = d_v = d_{model}/h$ in this paper.

After the multi-head attention network, there is a position-wise feedforward network with rectified linear unit (ReLU) activation:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \quad (4)$$

where $W_1 \in \mathbb{R}^{d_{model} \times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$, and the biases $b_1 \in \mathbb{R}^{d_{ff}}$, $b_2 \in \mathbb{R}^{d_{model}}$.

For decoder, there is a cross attention network between the multi-head self-attention network and feedforward network, whose structure is the same as the multi-head attention, except that the K and V come from the encoder while Q comes from the decoder. This allows the decoder to focus on different part of input speech frames in encoder for every decoding step [21]. Layer normalization and residual connection are applied before and after each module introduced above. To prevent the decoder from looking at subsequent text behind current position, a masking is applied to the future tokens.

2.2. Cross attention with alignment

In this section, we introduce our proposed Transformer model for speech recognition with modified cross attention that considers monotonicity between text output and speech input. The alignment information is beneficial for the decoder to focus on the relevant input speech frames, especially at lower layers where the model captures more local than global information [20]. This adjustment is carried out in both training and inference stages.

2.2.1. Alignment

Our objective is to find alignment between text output and speech input. Inspired by [22], which used recurrent neural network and proposed location-based attention mechanism computed by previous attention weights, we take advantage of current cross attention weights to locate the alignment position in the input and apply it on the transformer model. Given that a speech input and its corresponding text transcript should be most similar in the embedding space, we propose to take the position with the maximum cross attention weight as the input alignment for the current decoder input.

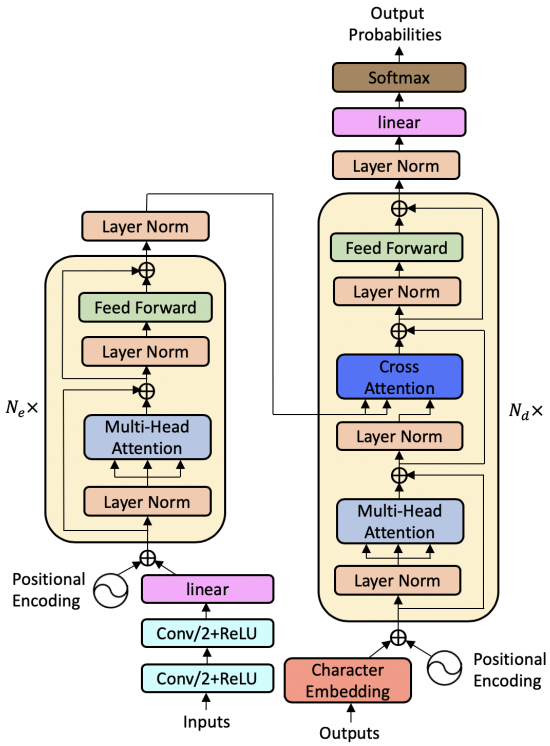

Figure 1: *Speech Transformer model architecture.*

For any text hidden vector y_i in the decoder, the cross attention weight α_{ij} to the encoder final output x_j in $X = (x_1, ..., x_n)$ is:

$$\alpha_{ij} = softmax(\frac{(y_i W^Q)(x_j W^K)^T}{\sqrt{d_k}}) \quad (5)$$

where $W^Q, W^K \in \mathbb{R}^{d_{model} \times d_k}$ are the linear transformation matrices. x_k aligns with y_i if:

$$\alpha_{ik} \geqslant \alpha_{ij} \ \ \forall j \in (1, n), j \neq k \quad (6)$$

i and k are the aligned output and input positions respectively, which will be used for cross attention computation.

2.2.2. Cross attention biasing

Attention biasing was proposed in [15] for improving the performance of self-attention acoustic model. Different from it, we propose to use attention biasing on the relevant part of encoder output under the cross attention framework. In particular, we add a Gaussian mask to the attention weights centered at the alignment position of encoder output which matches current decoder input most, which we name as soft attention biasing. The mask keeps the attention weight at the alignment position, and the weight is gradually diminished when moving away from the position. The attention weight is defined as:

$$\alpha_{ij} = softmax(\frac{(y_i W^Q)(x_j W^K)^T}{\sqrt{d_k}} + M_{ij}) \quad (7)$$

where M_{ij} is the attention mask defined in the following way:

$$M_{ij} = \frac{-(j-k)^2}{2\sigma^2} \quad (8)$$

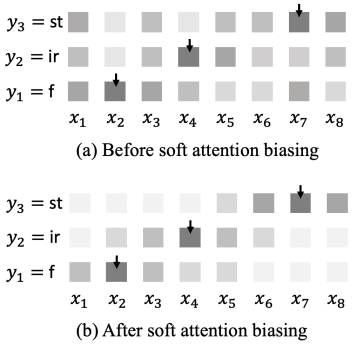

Figure 2: *Cross attention weight diagram before and after applying soft attention biasing. $X = (x_1, ..., x_8)$ are encoder final layer output vectors. Text outputs consist of sub-words "f ir st". Shade of gray in each square represents cross attention weight between each encoder and decoder vector. Darker shade means higher weight value. Squares that arrows point to represent the aligned positions identified using Eq. 6.*

where y_i aligns with x_k, and σ^2 is a learnable Gaussian variance parameter such that the aligned position has more attention weight value. Figure 2 shows the cross attention weights between text sub-words and encoder final layer output vectors before and after applying soft attention biasing. Besides soft attention biasing, we also try to mask all attention weights on the right hand side of alignment position (set them to 0) while keeping the left hand side attention weights, which we call it hard attention biasing. We will compare these two attention biasing techniques in our experiments.

2.2.3. Algorithmic details

Firstly, we *add n look-ahead frames* after alignment position. This is motivated by the benefit of adding some look-ahead frames after alignment position for location-aware attention in [18]. Eq. 8 in this case becomes:

$$M_{ij} = \frac{-(j-(k+n))^2}{2\sigma^2} \tag{9}$$

Secondly, we *apply cross attention biasing at lower layers of the decoder* as Eq. 7 to capture local information. This is motivated by [20] that the model captures more local than global information at lower layers. At higher layers where more global information is learned by the model, we remove the bias term and follow original cross attention model as Eq. 5 to attend to larger range of speech frames. In this way, our proposed method could fuse the local and global span of information. This is different from the hybrid global and local attention proposed in [23], which used encoder hidden states as a gating mechanism to integrate global and local attention in self-attention network.

2.3. Monotonic alignment regularization

Alignment positions between the output and input should be strictly monotonic in the input sequence for speech recognition. In order to regularize alignment using the technique above, we propose a regularizer term to achieve monotonicity. For an encoder output $X = (x_1, ..., x_n)$ and a text transcription embedding from the decoder $Y = (y_1, ..., y_m)$, we define the following misalignment loss as the regularization term:

$$loss_{misalign} = \sum_{l=1}^{m} \sigma(k_l - k_{l+1}) \tag{10}$$

Table 1: *WER results of end-to-end speech recognition models on LibriSpeech 100h*

Model	**Test_clean**	**Test_other**
Baseline transformer	12.0	29.7
Encoder-Decoder-Attention [24]	14.7	40.8
Encoder-Decoder-Attention (with data augmentation) [25]	15.1	-
LAS Model [26]	12.9	35.5

where σ is the sigmoid function and k_l is the position in X that y_l aligns to. In other words, we want the alignment of y_l to be in front (before) of the alignment of y_{l+1} in X. Under the CTC and attention hybrid multi-task learning framework, the proposed training criterion for improving the monotonic alignment between text output and speech input becomes:

$$loss = \alpha * loss_{ctc} + (1-\alpha) * loss_{att} + \beta * loss_{misalign} \tag{11}$$

$$loss_{ctc} = -lnP(y|x) \tag{12}$$

$$loss_{att} = \sum_{u} lnP(y_u|x, y_{1:u-1}) \tag{13}$$

where α is the ratio of CTC model loss in hybrid model.

3. Experiments

3.1. Experimental setup

We use ESPnet end-to-end speech recognition toolkit and LibriSpeech corpus for our experiments [27, 28]. The training dataset is 100 hour clean training data uttered by 251 speakers, and the development and test dataset are the default LibriSpeech development and test dataset, where each is around 5 hours and contains 2600 to 3000 utterances. LibriSpeech consists of 16kHz read English speech from audiobooks [28]. Input features are generated by 80-dimensional filterbanks with pitch on each frame, with a window size of 25ms shifted every 10ms. We exclude utterances longer than 3000 frames or 400 characters to keep memory manageable. We adopt a multi-task learning mechanism and joint decoding of CTC and attention [29], where output takes 30% of CTC output probability and 70% of attention output probability. β in Eq. 11 is chosen as 1.0 from preliminary experiments. The convolutional frontend before transformer encoder is two 2D convolutional neural network layers [13] with filter size (3,2), each followed by a ReLU activation. In the transformer model, the attention dimension d_{model} is 256, and feedforward network hidden state dimension d_{ff} is 2048, the number of attention heads is 4, the number of encoder layers N_e is 12, the number of decoder layers N_d is 6, the attention dropout rate is 0.0, the initial value of learning rate is 5.0, and the encoder and decoder dropout rate is 0.1. We use unigram sub-word algorithm with the vocabulary size capped to be 5000 [30].

3.2. Experimental results

3.2.1. Baseline system

We use the default settings of ESPnet transformer model as our baseline [27, 31]. The baseline model result is comparable with other published end-to-end speech recognition models [24, 25, 26], as shown in Table 1.

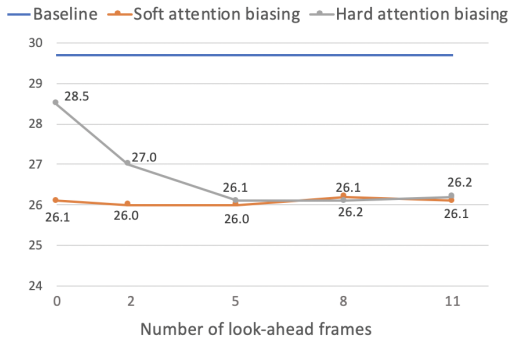

Figure 3: *WER results of different numbers of look-ahead frames on LibriSpeech Test_other set. Soft and hard attention biasing are applied on all decoder layers.*

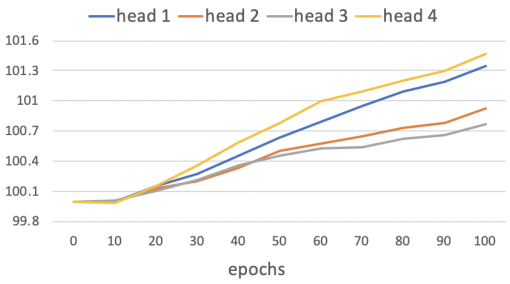

Figure 4: *Gaussian mask standard deviation for each of attention head at each training epoch.*

3.2.2. *The proposed system*

First, we explore the number of look-ahead frames included as mentioned earlier using two attention biasing techniques. The standard deviation σ in the Gaussian mask is initialized to 100 as in [15]. In both cases, we apply attention biasing on all the decoder layers. The results are shown in Figure 3. We can see that look-ahead frame does not affect much for soft attention biasing. This is because we employ a Gaussian mask with large learnable variance, so slight movement of Gaussian center position will have less impact. Figure 4 shows how the Gaussian mask standard deviation of each head evolves during training. However for hard attention biasing, since it depends entirely on the correctness of alignment position identified, careful tuning of number of look-ahead frames is required. Only when number of look-ahead frames is more than 5 in this case, hard attention biasing has comparable performance with soft attention biasing. To save the effort of tuning parameter, we employ soft attention biasing with 5 look-ahead frames in the following experiments.

Second, the number of decoder layers applying cross attention biasing is explored. The experimental results are listed in Table 2. We can see that using cross attention biasing at the lower layers (layer 1-3) can get slightly better results than at higher layers (layer 4-6). It is consistent with the previous results, that is, lower layers capture more local information, while higher layers capture more global information [20].

We investigate the effect of all three techniques used in our proposed method, that includes adding look-ahead frames, applying on lower layers of the decoder (layer 1-3), and monotonic alignment regularization. Table 3 lists the experimental results. We also test any two combination out of the three techniques by specifying missing which technique, e.g. *w/o look-ahead frames* refers to using the techniques of applying

Table 2: *WER results of applying cross attention biasing on different layers on LibriSpeech 100h*

Decoder layers applied	Test_clean	Test_other
baseline	12.0	29.7
layer 1-3	9.4	25.9
layer 4-6	9.5	26.2
layer 1-6	9.4	26.0

Table 3: *WER results of cross attention biasing and alignment regularization on LibriSpeech 100h*

Model	Test_clean	Test_other
baseline	12.0	29.7
our proposed method	**9.0**	**25.4**
w/o look-ahead frames	9.3	25.9
w/o applying on layer 1-3	9.4	26.0
w/o alignment regularization	9.4	25.9
hybrid global+local model [23]	9.4	25.9

on lower layers of the decoder and monotonic alignment regularization, and so on for the other two. It can be seen that these three techniques are equally important, as missing any one of them results in similar performance deterioration (row 3-5). For the last technique to encourage monotonic alignment between text output and speech input, we have tried to equally segment the input speech length based on output text length and then find alignment from corresponding speech frames to ensure monotonicity, but the results cannot surpass our alignment regularization method. Altogether, combination of all three techniques achieves our best result.

The last row of Table 3 lists the hybrid global and local attention approach [23]. Our model has 1.9%-4.3% relative WER reductions compared with [23]. Our model applies cross attention biasing on lower three decoder layers, and the rest decoder layers use standard cross attention. While in [23], it mixes two mechanism together with a gate derived from encoder hidden states. It is questionable on how much encoder hidden states can tell the importance of global versus local attention.

4. Conclusions

In this paper, we introduce an effective cross attention biasing technique through making use of cross attention weights. In order to constrain the monotonic alignment attribute between text output and speech input, we propose a regularizer term to achieve monotonicity. Under the cross attention framework, attention biasing limits the speech context range given alignment information. Experiments on LibriSpeech dataset denoted that our proposed model is effective. Comparing with the baseline system, there are 14.5%-25.0% relative WER reductions. Comparing with the hybrid global and local attention method, there are 1.9%-4.3% relative WER reductions. Obtaining output-input alignment in our proposed model relies on accurate encoding of speech input. In the future, we are interested to improve speech input encoding in self-attention network, and its integration with our cross attention with alignment method. Besides, we would explore identifying misalignment globally over entire utterances.

5. References

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

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

[3] 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.

[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] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[8] S. Ueno, H. Inaguma, M. Mimura, and T. Kawahara, "Acoustic-to-word attention-based model complemented with character-level ctc-based model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5804–5808.

[9] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association*, 2017, pp. 949–953.

[10] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[11] L. Deng, P. Kenny, M. Lennig, V. Gupta, F. Seitz, and P. Mermelstein, "Phonemic hidden markov models with continuous mixture output densities for large vocabulary word recognition," *IEEE Transactions on Signal Processing*, vol. 39, no. 7, pp. 1677–1681, 1991.

[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 *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[13] 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.

[14] K. J. Han, J. Huang, Y. Tang, X. He, and B. Zhou, "Multi-stride self-attention for speech recognition," in *INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, 2019, pp. 2788–2792.

[15] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. Waibel, "Self-attentional acoustic models," in *INTERSPEECH 2018 – 19th Annual Conference of the International Speech Communication Association*, 2018.

[16] 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 - ICML '06*. ACM Press, 2006, pp. 369–376.

[17] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association*, 2017, pp. 1298–1302.

[18] N. Moritz, T. Hori, and J. L. Roux, "Triggered attention for end-to-end speech recognition," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5666–5670.

[19] L. Dong and B. Xu, "CIF: Continuous integrate-and-fire for end-to-end speech recognition," *arXiv preprint arXiv:1905.11235*, 2020.

[20] A. Raganato and J. Tiedemann, "An analysis of encoder representations in transformer-based machine translation," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, 2018, pp. 287–297.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, 2015.

[22] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 EMNLP*, 2015, pp. 1412–1421.

[23] M. Xu, D. F. Wong, B. Yang, Y. Zhang, and L. S. Chao, "Leveraging local and global patterns for self-attention networks," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 3059–3075.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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, 2017.

[30] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018, pp. 66–75.

[31] 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 *INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, 2019, pp. 1408–1412.