The Conformer Encoder May Reverse the Time Dimension

Robin Schmitt, Albert Zeyer, Mohammad Zeineldeen, Ralf Schlüter and Hermann Ney
Human Language Technology and Pattern Recognition, Computer Science Department,
RWTH Aachen University, Aachen, Germany
AppTek GmbH, Aachen, Germany
Email: robin.schmitt1@rwth-aachen.de, {zeyer,zeineldeen,schlueter,ney}@cs.rwth-aachen.de

***Abstract*—We sometimes observe monotonically *decreasing* cross-attention weights in our Conformer-based global attention-based encoder-decoder (AED) models, negatively affecting performance compared to monotonically increasing attention weights. Further investigation shows that the Conformer encoder reverses the sequence in the time dimension. We analyze the initial behavior of the decoder cross-attention mechanism and find that it encourages the Conformer encoder self-attention to build a connection between the initial frames and all other informative frames. Furthermore, we show that, at some point in training, the self-attention module of the Conformer starts dominating the output over the preceding feed-forward module, which then only allows the reversed information to pass through. We propose methods and ideas of how this flipping can be avoided and investigate a novel method to obtain label-frame-position alignments by using the gradients of the label log probabilities w.r.t. the encoder input frames.**

***Index Terms*—Conformer encoder, AED models**

I. Introduction & related work

The Conformer encoder [1] is widely used in automatic speech recognition (ASR) systems. It combines self-attention and convolutional layers to model both long-range and local dependencies in the input sequence and outperforms [1], [2] the Transformer encoder [3]. In the attention-based encoder-decoder (AED) framework, the decoder autoregressively produces an output sequence while attending to the whole encoder output at each step [4], [5].

We observe that the Conformer encoder of an AED model sometimes *reverses the time dimension* of the input sequence, as the cross-attention weights show (Figure 1). This work investigates why this phenomenon occurs and how it can be avoided.

Furthermore, we propose a novel way to obtain label-frame-position alignments (for each output label, the start/end time frames in the input) by leveraging the gradients of the label log probabilities w.r.t. the encoder input frames. We use methods related to saliency maps and other gradient-based attribution methods such as the ones studied in [6], [7]. Moreover, [8], [9] shows how gradients could be used to better interpret the behavior of a model.

II. AED Model

Our baseline is the standard AED model adapted for ASR [4], [5]. The encoder consists of a convolutional frontend, which downsamples the sequence $x_1^{T'}$ of 10ms frames into a sequence ${h_0}_1^T$ of 60ms frames, followed by a stack of $N = 12$ Conformer blocks, which further process the downsampled sequence resulting in the encoder output h_1^T:

$$
\begin{aligned}
{h_0}_1^T &= \mathrm{ConvFrontend}(x_1^{T'}), \\
{h_i}_1^T &= \mathrm{ConformerBlock}_i({h_{i-1}}_1^T), \quad h = h_N, i = 1, \dots, N.
\end{aligned}
$$

The probability of the output label sequence a_1^S given the encoder output h_1^T is defined as

$$p(a_1^S \mid h_1^T) = \prod_{s=1}^{S} p(a_s \mid a_1^{s-1}, h_1^T).$$

At each step s, the decoder autoregressively predicts the next label a_s by attending over the whole encoder output h_1^T. We define $a_S = \texttt{EOS}$ as the end-of-sequence token, which implicitly models the probability of the sequence length. We use an LSTM [10] decoder with single-headed MLP cross-attention [11] following our earlier setup [12]. We use BPE subword label units [13] (1k and 10k vocab. sizes).

We train our model using the standard label-wise cross-entropy (CE) criterion $L = -\log p(\overline{a}_1^S \mid h_1^T)$ using the target transcriptions $\overline{a}_1^S$. In the experiments, where we observe the flipping of cross-attention weights, there is no further loss, but we also tested to add CTC [14] as an auxiliary loss [15].

We use label-synchronous beam search for recognition.

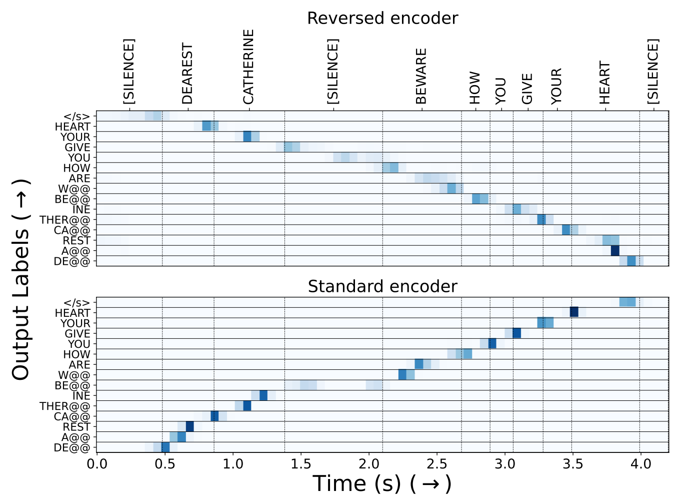

Fig. 1: Cross-attention weights of a model with reversed encoder vs. standard encoder.

TABLE I: Comparing the baseline in this paper – Conformer AED with BPE 1k without CTC aux. loss – vs related models. Results on LibriSpeech, without external language model.

AED Model	Label Units	CTC aux. loss	Dis. self-att. 1st ep.	Flip. enc.	WER [%] dev clean	dev other	test clean	test other
E-Branchf.-Trafo [25]	BPE 5k	Yes	No	No	2.0	4.6	2.1	4.6
Conformer-LSTM	BPE1k	No	No	Yes	2.8	6.8	3.0	6.6
				No	2.3	5.8	2.5	5.8
		Yes			2.2	5.6	2.5	5.6
		No	Yes		2.2	5.4	2.5	5.7
		Yes			2.3	5.7	2.5	5.5
	BPE10k	No	No		2.6	6.1	2.8	6.0
		Yes			2.3	5.6	2.5	5.7

III. Experimental Setup

Our experimental setup follows the global AED baseline from [16]. We perform experiments on the LibriSpeech 960h [17] corpus using the RETURNN framework [18] based on PyTorch [19]. Our pipeline is managed by Sisyphus [20]. All models are trained for 100 epochs with the AdamW optimizer [21], using on-the-fly speed perturbation and SpecAugment [22]. As hardware, we either use a single Nvidia A10 GPU or 4x Nvidia 1080 GTX GPUs in parallel with parameter syncing every 100 batches [23], [24]. All the code to reproduce our experiments is published[1].

Table I compares the word error rate (WERs) of our baseline model to related models. BPE 1k without CTC aux. loss is the base configuration of all experiments. Flipping has a negative effect on the WER while our measures against flipping have a positive effect. Notably, our flipped models show significant variation in WER (6.8-21.6% on dev-other) when using different random seeds, with only the best result included in the table.

IV. Analysis

A. Initial Development of Cross-Attention Weights

In all of our experiments, the decoder cross-attention initially only attends to the first few frames of the encoder output (Figure 2). We hypothesize that the choice of these frames is not due to their usefulness for predicting labels but rather because they have distinct features, such as convolutional zero padding on the sequence boundaries and/or initial silence, which makes it simple to attend to them. The first frame is probably easier to recognize than the last, as the last frame can have different padding due to the batching, and also the sequences on LibriSpeech have slightly more silence at the beginning (260 ms vs. 230 ms on average).

We conducted an experiment where we force the model to only attend to the center frame by setting the attention weight of this frame to one and all other weights to zero. The initial CE losses of these models are almost identical (6.30 vs 6.29 after 1 epoch), proving that the choice of frame for the initial cross-attention weights is not important for label prediction early in training.

To see how the model utilizes frames from individual encoder layers, we examine the gradients of the target label $\overline{a}_s$

[1] https://github.com/rwth-i6/returnn-experiments/tree/master/2024-flipped-conformer

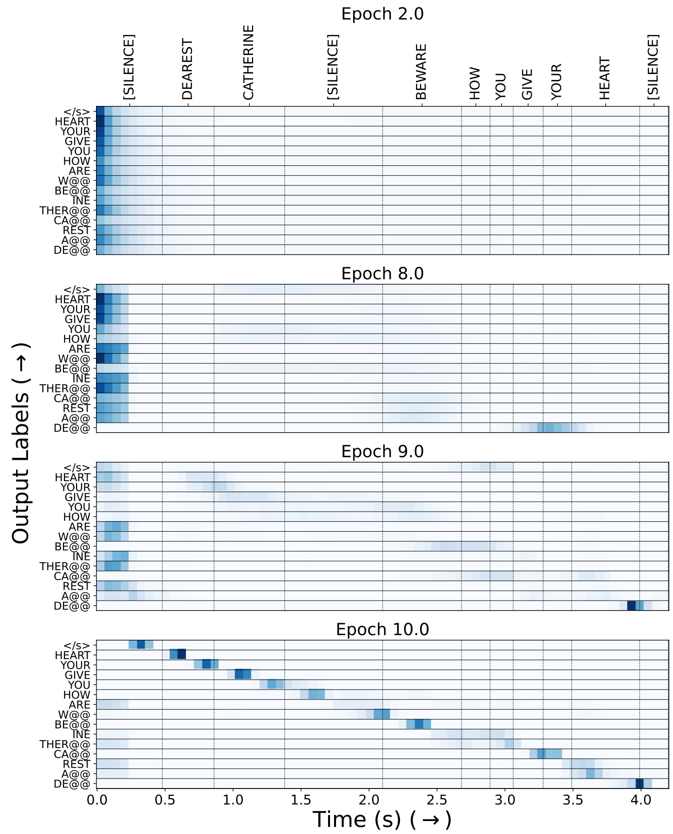

Fig. 2: Development of cross-attention weights into the flipping behavior over the initial training epochs.

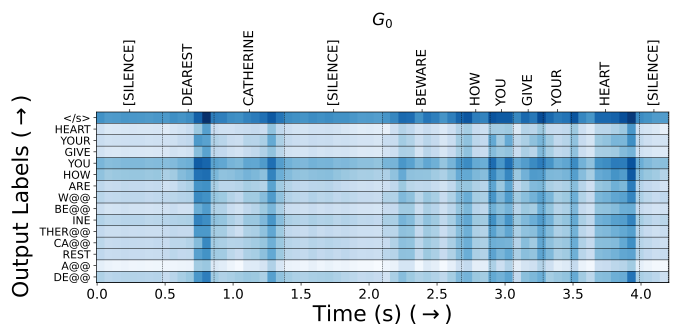

Fig. 3: Showing G_0, i.e. the logarithm of the L_2 norm of the gradients of the target label log probabilities w.r.t. first Conformer block inputs very early in training (after 2 epochs).

log probabilities w.r.t. the corresponding encoder layer input or output h_i or the convolutional frontend input $h_{-1} = x$. This gives us a D-dimensional vector per target label position s and frame t, which we reduce to a scalar by taking the logarithm of its L_2 norm:

$$G_{i_{s,t}} = \log \left\| \nabla_{h_{it}} \log p(\overline{a}_s \mid \overline{a}_1^{s-1}, h_1^T) \right\|_2 . \qquad (1)$$

Early in training (after 2 epochs), G_0 (Figure 3) shows that the gradients are more focused on individual label frames and less focused on the silence frames. For some labels, we can see that, in this early stage in training, the gradients over all input frames are stronger, meaning the encoder output is more important for those labels than for others. It is specifically more pronounced for the labels "You" and `EOS`.

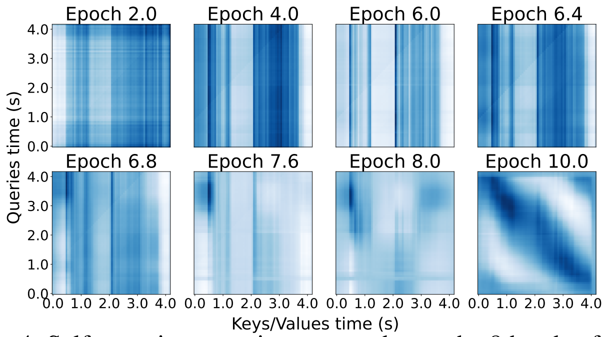

Fig. 4: Self-attention energies averaged over the 8 heads of the 10th Conformer block for initial epochs. After this, all further layers are flipped.

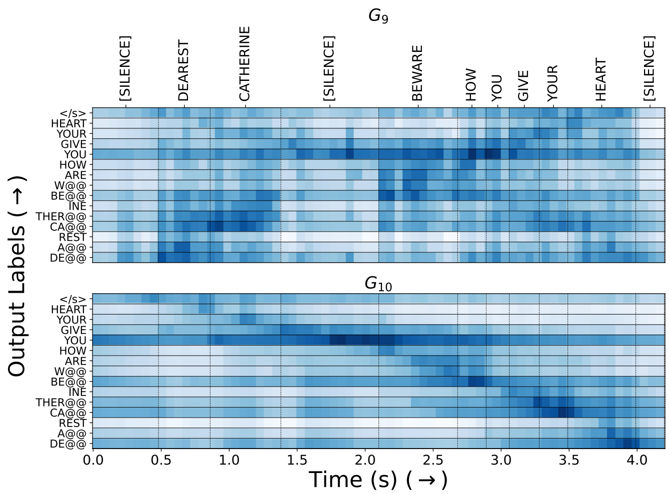

Fig. 5: Gradients G_9 and G_{10} w.r.t. the output of blocks 9 and 10 after 12 epochs. For G_9, we have the crossing of information from the residual and the self-attention. In G_{10}, only the flipped information is left.

B. How is Time Reversal possible?

The flipping can only occur in the self-attention (Figure 4). But what about the residual connection? In every Conformer block, there is a final layernorm layer, so there is no direct residual path from the input to the output of the encoder. There is one Conformer block where the time reversal occurs. In the flipping block, the self-attention module has by far the largest activations in magnitude (mean L_2 norm 60 for some example sequence), so that the residual connection (norm 23) from the input of this block and from the first feed-forward module do not have much effect anymore. The remaining convolution module and second feed-forward module also do not add much (norm 13 and 15 respectively). The final layernorm then removes all the original frame-wise information and only the reverse order is kept.

Flipping occurs in Conformer block 10 and not in the other blocks (Figure 5). The gradients G_9 (Figure 5) show that there is still some information remaining from the inner residual connection in block 9.

We do not expect that such flipping is easy to perform by an encoder where there is a residual connection from input to output like in the standard Transformer [3].

C. Reasons for Time Reversal

1) The cross-attention initially only attends to the first few encoder frames for all decoder frames (Figure 2, epoch 2) as discussed before (Section IV-A).

2) The decoder acts initially like a language model and can learn independently from the encoder. To slightly improve the prediction perplexity, having some information of the encoder is useful, and that is where it uses the fixed cross-attention to the first frame. The first encoder frame attends globally to more informative frames of lower layers (Figure 4, epoch 2 to 6.4) as this is most useful at this stage to collect global information from all labels. Attending globally to multiple frames is also more informative than just first label or another single label for predicting the whole sequence.

3) To further improve the sequence prediction perplexity, the cross-attention learns to focus on another frame, which happens to be some random frame towards the end (Figure 2, epoch 8). The next most easily usable information is to know about the first label in the sequence[2]. Thus this encoder frame uses self-attention to attend to the frames of the first label (Figure 4, epoch 6.8 to 8), where the first label is originally located in lower layers.

4) The next labels follow, one after the other. For the cross-attention, it is easier to choose some position right next to the previous position[3]. So this will lead to the flipping (Figure 2 and Figure 4, epoch 8 to 10).

Vocabulary and average sequence length influence those training dynamics. Specifically, skipping sequences longer than 75 labels during training prevents flipping, both for BPE1k and BPE10k. For BPE1k, the output sequences are naturally longer.

The training dynamics are stochastic and depend on the random initialization. Across 13 experiments with different random seeds, the flipping happens in 12 cases (for BPE1k). In 2 out of the 12 flipping cases, we also observed some shuffling of segments.

D. Measures to Avoid Time Reversal

1) Use CTC auxiliary loss [15]: CTC enforces monotonic alignments between input frames and output labels, so the encoder output can not be flipped. We never observed flipping in experiments using CTC auxiliary loss. This is commonly used (by default in ESPnet [26] and many RETURNN setups [12], [16]), which is why the flipping was maybe not observed before.

2) Disabling self-attention in the beginning: Flipping occurs in the self-attention. We can fix the self-attention weights to be the identity matrix for the first epoch and then only later use learned attention weights. This means that we only use the linear transformation for the values in the self-attention module. This experiment shows faster convergence (after 12

[2]The last label is probably more difficult for the decoder to handle, because it must also learn when that last label occurs. It's very easy to know that it must predict the first label.

[3]Due to positional information, when it has attended close to it in the previous decoder frame, it should be easy to identify the next closest label in the encoder frames.

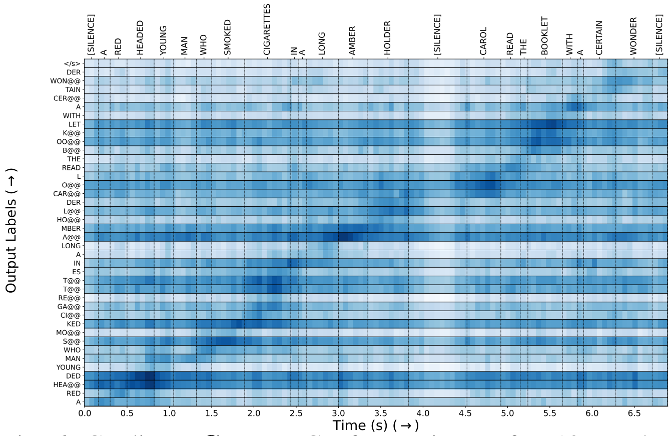

Fig. 6: Gradients G_0 w.r.t. Conformer input after 12 epochs. Alignment path is visible.

epochs: baseline reaches 0.83 CE loss, baseline with CTC aux loss reaches 0.56 CE, disabling self-attention reaches 0.45 CE), and also no flipping occurs.

3) Hard attention on center frame: We argued before that the initial focused cross-attention to the first frame might lead to the flipping. We tried forcing the cross-attention weights to the center of the encoder output sequence for the first full epoch. In this experiment, no flipping did occur.[4]

V. Alignments from Model Input Gradients

As shown in Figure 6, the *gradients of the log label probabilities w.r.t. the Conformer input* G_0 show an alignment between output labels and input frames. We were asking the question: Can we use those gradients to estimate an alignment, i.e. the best alignment path (boundaries/positions of each label and word)? This has been done before using the attention weights [27] but that can be problematic due to multiple cross-attentions (e.g. Transformer with multiple layers and heads) or because the encoder shifts or transforms the input, as our work shows here. When the model has reasonable performance, the gradients w.r.t. the model input cannot have those artifacts like flipping or shifting.

We use either G_0 (Equation (1)) to get an alignment with 60ms frame shift or G_{-1} (before the conv. frontend) with 10ms frame shift. We allow any number of ϵ (blank) labels between any of the real labels, we allow the real label to be repeated multiple times over time, and we exclude the final `EOS` label. This is very similar to the CTC label topology except that we do not enforce an ϵ between two equal labels. We search for an allowed state sequences $r_1^T : a_1^S$ for state indices $r_t \in \{1, \ldots, 2 \cdot S - 1\}$ corresponding to states $Y = (\epsilon, 1, \epsilon, 2, \ldots, S-1, \epsilon)$ which maximizes

$$\mathrm{GradScore}(r_1^T) = \sum_{t=1}^{T} \mathrm{GradScore}(r_t) \quad (2)$$

$$\mathrm{GradScore}(r_t) = \begin{cases} \log \mathrm{softmax}_{\bar{t}}(G_i)_{Y_{r_t}, t}, & Y_{r_t} \neq \epsilon, \\ \gamma_\epsilon, & Y_{r_t} = \epsilon \end{cases} \quad (3)$$

[4]But we might need more experiments to really be sure.

TABLE II: Alignment quality in terms of time-stamp-error (TSE) for word left/right boundaries and center positions against reference GMM alignment on a random 10h subset of the training data of LibriSpeech. *: On the whole training data, and the computation is different: more consistent alignment due to same feature extraction as the GMM, while our models here use a different feature extraction.

Best Path Scores	Model	Train Phase	Flip	Frame Shift [ms]	Label Units	TSE [ms]	
						Left/ Right	Center
Probs.	CTC [30]	full	no	40	Phonemes	38*	-
Probs.	CTC	full	no	60	BPE1k	83	61
					BPE10k	312	306
Grads.	AED	early	no	10	BPE1k	56	43
				60		61	50
			yes	10		72	53
				60		75	61

for some fixed blank score γ_ϵ which is a hyperparameter[5]. The best r_1^T can be found via dynamic programming. We obtain the final alignment label sequence y_1^T with $y_t = \begin{cases} a_{Y_{r_t}}, & Y_{r_t} \neq \epsilon, \\ \epsilon, & Y_{r_t} = \epsilon \end{cases}$.

We measure the time-stamp-error (TSE) [28]–[30] of word boundaries, i.e. the mean absolute distance (in milliseconds) of word start and end positions against a reference GMM alignment, irrespective of the silence. Additionally, we also compute the TSE w.r.t. the word center positions, which might be a better metric for peaky alignments like CTC [31]. We summarize our findings in Table II. Our method still performs worse in terms of TSE compared to a well tuned phoneme-level CTC model from earlier work [30], however we improve over all our BPE-based CTC alignments by far. We also see that our method still works even when the encoder flips the sequence.

VI. Conclusions

In this work, we have shown that the Conformer encoder of an attention-based encoder-decoder model is able to reverse the time dimension of the input sequence. The flipping is favored heavily due to the Conformer structure with its missing residual connections, not using CTC (or other monotonicity enforcing methods), and having longer sequence lengths on the decoder side. Furthermore, we proposed several methods to avoid this flipping such as disabling the self-attention in the beginning, which also improves convergence.

We also showed that the gradients of the label log probabilities w.r.t. the encoder input frames can be used to obtain alignments, even early in training, even when the encoder reverses the sequence, and its time-stamp-errors are even better than a normal CTC forced alignment.

VII. Acknowledgments

This work was partially supported by the project RESCALE within the program *AI Lighthouse Projects for the Environment, Climate, Nature and Resources* funded by the Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV), funding ID: 67KI32006A.

[5]We use $\gamma_\epsilon = -4$ for the 60ms shift (G_0) and $\gamma_\epsilon = -6$ for the 10ms shift (G_{-1}).

References

[1] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-3015

[2] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, J. Shi, S. Watanabe, K. Wei, W. Zhang, and Y. Zhang, "Recent developments on ESPnet toolkit boosted by Conformer," 2020. [Online]. Available: https://arxiv.org/abs/2010.13956

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 6000–6010.

[4] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," Preprint arXiv:1506.07503, 2015. [Online]. Available: http://arxiv.org/abs/1506.07503

[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 *ICASSP*, 2016.

[6] M. Ancona, E. Ceolini, A. C. Öztireli, and M. H. Gross, "A unified view of gradient-based attribution methods for deep neural networks," Preprint arXiv:1711.06104v1, 2017. [Online]. Available: http://arxiv.org/abs/1711.06104v1

[7] A. Prasad and P. Jyothi, "How accents confound: Probing for accent information in end-to-end speech recognition systems," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 3739–3753. [Online]. Available: https://aclanthology.org/2020.acl-main.345

[8] S. Serrano and N. A. Smith, "Is attention interpretable?" in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 2931–2951. [Online]. Available: https://doi.org/10.18653/v1/p19-1282

[9] Y. Hechtlinger, "Interpretation of prediction models using the input gradient," Preprint arXiv:1611.07634, 2016. [Online]. Available: http://arxiv.org/abs/1611.07634

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, Eds. The Association for Computational Linguistics, 2015, pp. 1412–1421. [Online]. Available: https://doi.org/10.18653/v1/d15-1166

[12] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *19th Annual Conference of the International Speech Communication Association, Interspeech 2018, Hyderabad, India, September 2-6, 2018*, B. Yegnanarayana, Ed. ISCA, 2018, pp. 7–11. [Online]. Available: https://doi.org/10.21437/Interspeech.2018-1616

[13] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. [Online]. Available: https://doi.org/10.18653/v1/p16-1162

[14] 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.

[15] 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.

[16] M. Zeineldeen, A. Zeyer, R. Schlüter, and H. Ney, "Chunked attention-based encoder-decoder model for streaming speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*. IEEE, 2024, pp. 11 331–11 335. [Online]. Available: https://doi.org/10.1109/ICASSP48485.2024.10446035

[17] 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)*, 2015, pp. 5206–5210.

[18] A. Zeyer, T. Alkhouli, and H. Ney, "RETURNN as a generic flexible neural toolkit with application to translation and speech recognition," in *Annual Meeting of the Assoc. for Computational Linguistics*, Melbourne, Australia, Jul. 2018.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," Preprint arXiv:1912.01703, 2019. [Online]. Available: http://arxiv.org/abs/1912.01703

[20] J. Peter, E. Beck, and H. Ney, "Sisyphus, a workflow manager designed for machine translation and automatic speech recognition," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, E. Blanco and W. Lu, Eds. Association for Computational Linguistics, 2018, pp. 84–89. [Online]. Available: https://doi.org/10.18653/v1/d18-2015

[21] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017. [Online]. Available: http://arxiv.org/abs/1711.05101

[22] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[23] R. McDonald, K. Hall, and G. Mann, "Distributed Training Strategies for the Structured Perceptron," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, R. Kaplan, J. Burstein, M. Harper, and G. Penn, Eds. Los Angeles, California: Association for Computational Linguistics, Jun. 2010, pp. 456–464.

[24] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 215–219.

[25] K. Kim, F. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, "E-branchformer: Branchformer with enhanced merging for speech recognition," in *IEEE Spoken Language Technology Workshop, SLT 2022, Doha, Qatar, January 9-12, 2023*. IEEE, 2022, pp. 84–91. [Online]. Available: https://doi.org/10.1109/SLT54892.2023.10022656

[26] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[27] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 2023, pp. 28 492–28 518. [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html

[28] X. Zhang, V. Manohar, D. Zhang, F. Zhang, Y. Shi, N. Singhal, J. Chan, F. Peng, Y. Saraf, and M. Seltzer, "On lattice-free boosted MMI training of HMM and CTC-based full-context ASR models," in *ASRU*, 2021.

[29] T. Raissi, W. Zhou, S. Berger, R. Schlüter, and H. Ney, "HMM vs. CTC for Automatic Speech Recognition: Comparison Based on Full-Sum Training from Scratch," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, Jan. 2023, pp. 287–294.

[30] T. Raissi, C. Lüscher, S. Berger, R. Schlüter, and H. Ney, "Investigating the effect of label topology and training criterion on asr performance and alignment quality," in *Interspeech*, Kos, Greece, Sep. 2024, preprint Arxiv:2407.11641.

[31] A. Zeyer, R. Schlüter, and H. Ney, "Why does CTC result in peaky behavior?" Preprint arXiv:2105.14849, May 2021. [Online]. Available: http://arxiv.org/abs/2105.14849