INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

StableEmit: Selection Probability Discount for Reducing Emission Latency of Streaming Monotonic Attention ASR

Hirofumi Inaguma, Tatsuya Kawahara

Graduate School of Informatics, Kyoto University, Kyoto, Japan
{inaguma,kawahara}@sap.ist.i.kyoto-u.ac.jp

Abstract

While attention-based encoder-decoder (AED) models have been successfully extended to the online variants for streaming automatic speech recognition (ASR), such as monotonic chunkwise attention (MoChA), the models still have a large label emission latency because of the unconstrained end-to-end training objective. Previous works tackled this problem by leveraging alignment information to control the timing to emit tokens during training. In this work, we propose a simple *alignment-free* regularization method, *StableEmit*, to encourage MoChA to emit tokens earlier. StableEmit discounts the selection probabilities in hard monotonic attention for token boundary detection by a constant factor and regularizes them to recover the total attention mass during training. As a result, the scale of the selection probabilities is increased, and the values can reach a threshold for token emission earlier, leading to a reduction of emission latency and deletion errors. Moreover, StableEmit can be combined with methods that constraint alignments to further improve the accuracy and latency. Experimental evaluations with LSTM and Conformer encoders demonstrate that StableEmit significantly reduces the recognition errors and the emission latency simultaneously. We also show that the use of alignment information is complementary in both metrics.
Index Terms: Streaming automatic speech recognition, monotonic chunkwise attention, emission latency

1. Introduction

End-to-end (E2E) automatic speech recognition (ASR) models have been recently studied to streamline the complicated pipelines in conventional hybrid systems. Online streaming is one of the most important tasks for real-world applications. While frame-synchronous models such as connectionist temporal classification (CTC) [1] and RNN transducer (RNN-T) [2] are natural choices for the streaming purpose, emerging progress of streaming attention-based encoder-decoder (AED) models, such as monotonic chunkwise attention (MoChA) [3], make them good alternatives as well [4, 5].

Apart from the decoder topology, all streaming E2E models have a large emission latency because the alignment-unconstrained E2E training objective encourages the model to see as many future observations as possible. To tackle this problem, various methods have been proposed; alignment path restriction [6–8], encoder pre-training with frame-level supervision [7–9], and a new training objective for direct latency minimization [7, 10–13]. Among them, FastEmit [11] successfully reduced the emission latency of RNN-T by modifying the loss to prioritize the generation of non-blank labels over blank labels *without external alignment*. However, the method is specific to RNN-T and cannot be applied to MoChA directly. This is because MoChA detects token boundaries with a Bernoulli random variable in the internal attention module while RNN-T treats all labels, including a blank label, equally in the output softmax layer.

In this work, we propose a straightforward alignment-free regularization method, *StableEmit*, to reduce the emission latency of MoChA. StableEmit discounts selection probabilities for token boundary detection in MoChA with a constant factor during training. Simultaneously, we encourage the expected attention weights calculated from the modified selection probabilities, to sum up to one to recover the attention mass (*quantity regularization* [10]). Therefore, the confidence score for token emission is enhanced and reaches a threshold to emit a token earlier at test time. This also reduces deletion errors, which stabilizes the search.

Moreover, it is possible to combine StableEmit with alignment regularization methods such as delay constrained training (DeCoT) [7] and CTC-synchronous training (CTC-ST) [10]. We further propose DeCoT-CTC, a new alignment path restriction method combining these ideas.

Experimental evaluations with Long Short-Term Memory (LSTM) and causal Conformer [14] encoders show that the proposed method significantly reduces the deletion errors and the emission latency of MoChA. We also demonstrate that the combination with the alignment regularization methods has a complementary effect, leading to further improvements in the accuracy and latency. The best MoChA model shows comparable performance to RNN-T on long-form speech utterances, which has been difficult to achieve so far.

2. Background

2.1. Monotonic chunkwise attention (MoChA)

MoChA is a linear-time attention model on the basis of hard monotonic attention (HMA) [15]. To compensate for the performance degradation due to the hard attention to a single encoder output, MoChA introduces an additional soft attention module restricted to a local chunk of w frames from the boundary detected by HMA. The test-time decoding complexity is linear, in which the discrete decision $z_{i,j} \in \{0, 1\}$ to generate the i-th token on the j-th encoder output is sampled from a Bernoulli random variable parameterized by a selection probability $p_{i,j}$.

Because this procedure is not differentiable, the expected alignment probabilities $\alpha_{i,j}$ is calculated by summing all possible alignment probabilities during training as

$$\begin{aligned}\alpha_{i,j} &= p_{i,j} \sum_{k=1}^{j} \left(\alpha_{i-1,k} \prod_{l=k}^{j-1} (1 - p_{i,l}) \right) \\ &= p_{i,j} \left((1 - p_{i,j-1}) \frac{\alpha_{i,j-1}}{p_{i,j-1}} + \alpha_{i-1,j} \right). \end{aligned} \quad (1)$$

While Eq. (1) requires the iterative update over j, there is an effective implementation which parallelizes the calculation by

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-1110

using the cumulative sum and product operations, denoted respectively as $\texttt{cumsum}$ and $\texttt{cumprod}$, as

$$\boldsymbol{\alpha}_{i,:} = \boldsymbol{p}_i \cdot \texttt{cumprod}(1 - \boldsymbol{p}_{i,:}) \cdot \texttt{cumsum}\left(\frac{\boldsymbol{\alpha}_{i-1,:}}{\texttt{cumprod}(1 - \boldsymbol{p}_{i,:})}\right). \quad (2)$$

The chunkwise attention is estimated with $\alpha_{i,j}$ at training time and is calculated from the hard boundary at test time. The interested readers are referred to [3, 15] for more details.

2.2. Quantity regularization

An exponential decay of $\alpha_{i,j}$ due to the recurrence in Eq. (1) enlarges the mismatch between training and testing behaviors, which is the main cause of the poor performance of the original MoChA [4, 16, 17]. To mitigate this problem, quantity regularization [7] was proposed by introducing a quantity loss $\mathcal{L}_{\text{qua}}$, which makes the expectation of the total number of boundaries closer to the output sequence length U as $\mathcal{L}_{\text{qua}} = |U - \sum_{i,j} \alpha_{i,j}|$.

2.3. Alignment path restriction

One of the effective methods to reduce the emission latency of MoChA is to ignore paths in Eq. (1) by referring to external alignment information extracted from the hybrid system [7]. Given reference boundaries $\mathbf{b}^{\text{ref}} = (\mathrm{b}_1^{\text{ref}}, \cdots, \mathrm{b}_U^{\text{ref}})$ estimated from the word alignment, delay constrained training (DoCoT) [7] masks out inappropriate paths based on

$$\alpha_{i,j} = \begin{cases} p_{i,j}\left((1 - p_{i,j-1})\frac{\alpha_{i,j-1}}{p_{i,j-1}} + \alpha_{i-1,j}\right) & (j \leq \mathrm{b}_i^{\text{ref}} + \delta) \\ 0 & (\text{otherwise}), \end{cases}$$

where δ is a hyperparameter to control the delay. To recover the reduced scale of $\alpha_{i,j}$, it is essential to use quantity regularization [7].

Moreover, it is also possible to use CTC alignment to estimate $\mathbf{b}^{\text{ref}}$. We refer to this method as *DeCoT-CTC* and note that it is newly investigated in this work. Although the CTC alignment is not guaranteed to be accurate, it will be effective because the model still has some freedom to learn the optimal timing to emit tokens, unlike point-wise boundary restriction in the expected latency loss described in Section 2.4. Moreover, the CTC alignment can be obtained without frame-level alignment supervision. During training, we use the most probable CTC alignment as in [10].

2.4. Expected latency loss

Another strategy to reduce the latency is to use boundary supervision by minimizing the expected latency loss [7, 10] as

$$\mathrm{b}_i^{\text{mocha}} = \sum_{j=1}^{T'} j \cdot \alpha_{i,j},$$

$$\mathcal{L}_{\text{latency}} = \frac{1}{U} \sum_{i=1}^{U} |\mathrm{b}_i^{\text{ref}} - \mathrm{b}_i^{\text{mocha}}|,$$

where $\mathrm{b}_i^{\text{mocha}}$ is the expected boundary index for the i-th token, and T' is the time length of encoder outputs. When using the alignment from a hybrid system, training becomes minimum latency training (MinLT) [7]. When using the CTC alignment, it becomes CTC-synchronous training (CTC-ST) [10]. We have shown that CTC-ST outperforms MinLT in the accuracy and latency [4] and therefore focus on the former in this work.

Table 1: *Summary of alignment regularization for MoChA*

		Source of alignment	
		CTC	Hybrid ASR
Training	Expected latency loss	CTC-ST [10]	MinLT [7]
	Alignment path restriction	DeCoT-CTC (this work)	DeCoT [7]

2.5. Total training objective

To summarize the above methods, the generalized training objective of MoChA is formulated as follows:

$$\mathcal{L}_{\text{mocha}}^{\text{total}} = (1 - \lambda_{\text{ctc}})\mathcal{L}_{\text{mocha}} + \lambda_{\text{ctc}}\mathcal{L}_{\text{ctc}} + \lambda_{\text{latency}}\mathcal{L}_{\text{latency}} + \lambda_{\text{qua}}\mathcal{L}_{\text{qua}}, \quad (3)$$

where λ_* is a corresponding task weight. When $\lambda_{\text{latency}} > 0$, λ_{qua} is set to 0.0 in this work [10]. We summarize the above alignment regularization methods in Table 1.

3. StableEmit

In this section, we propose *StableEmit*, a simple but effective emission latency regularization method for MoChA without external alignment information.

3.1. Motivation

While the methods in Section 2.3 and Section 2.4 can reduce the emission latency of MoChA, they require alignment information, although the CTC alignment can be obtained easily. Moreover, at test time, token boundaries are detected when and only when $p_{i,j}$ surpasses a threshold $\tau = 0.5$ [15].[1] Therefore, if the confidence of boundary prediction in some speech frames is below τ, the corresponding token cannot be generated because it does not appear in the subsequent search process. As a result, this leads to increasing deletion errors, especially for long-form speech data.

3.2. Formulation

To solve this problem, we propose *StableEmit*, which discounts $p_{i,j}$ *during training only* to detect token boundaries earlier with high confidence at test time. Specifically, we introduce a constant discount factor λ_{se} $(0 \leq \lambda_{\text{se}} < 1)$ and multiply $p_{i,j}$ by $1 - \lambda_{\text{se}}$. Therefore, Eq. (1) is reformulated as

$$\begin{aligned} p'_{i,j} &= (1 - \lambda_{\text{se}}) \cdot p_{i,j}, \\ \alpha_{i,j} &= p'_{i,j}\left((1 - p'_{i,j-1})\frac{\alpha_{i,j-1}}{p'_{i,j-1}} + \alpha_{i-1,j}\right). \end{aligned} \quad (4)$$

Accordingly, Eq. (2) is rewritten as

$$\boldsymbol{\alpha}_{i,:} = \boldsymbol{p}'_{i,:} \cdot \texttt{cumprod}(1 - \boldsymbol{p}'_{i,:}) \cdot \texttt{cumsum}\left(\frac{\boldsymbol{\alpha}_{i-1,:}}{\texttt{cumprod}(1 - \boldsymbol{p}'_{i,:})}\right).$$

Similarly to FastEmit, StableEmit is also a *sequence-level* regularization because the discount of $p_{i,j}$ at every grid (i, j) influences $\alpha_{i,j}$ globally. As the discount decreases $\sum_j \alpha_{i,j}$, we further need to use the quantity regularization to recover it. With this technique, $p_{i,j}$ is optimized to detect token boundaries with

[1] Setting $\tau = 0.5$ is reasonable when assuming $p_{i,j}$ is well discretized after the training.

high confidence because it is encouraged to compensate the discount so that $\sum_j \alpha_{i,j}$ gets closer to 1. In other words, the reasonable value of τ increases from 0.5 to $0.5/(1-\lambda_{\mathrm{se}})$ through the training. However, at test time, we keep $\tau = 0.5$ as in the original formulation, unless otherwise noted. This is effective for reducing the emission latency because $p_{i,j}$ can reach τ earlier in terms of j if it is trained to recover the discount properly.

Since StableEmit just modifies $p_{i,j}$ during training, it can be combined with alignment regularization methods that manipulate $\alpha_{i,j}$ such as DeCoT and CTC-ST. The effect of combination will be described in Section 4.

4. Experimental evaluations

4.1. Experimental setup

We used the TEDLIUM release v2 (TEDLIUM2) [18], a 210-hour English lecture speech corpus, and the evaluation sets include long-form utterances (40/30 seconds in the *dev*/*test* sets, respectively). For the GPU memory capacity, however, we removed utterances longer than 16 and 20 seconds in the training data for LSTM and Conformer models, respectively. We extracted 80-channel log-mel filterbank coefficients computed with a 25-ms window that was shifted every 10ms with Kaldi [19]. We applied three-fold speed perturbation [20] and SpecAugment [21]. The vocabularies were constructed by the byte pair encoding (BPE) algorithm [22] with 10k and 1k units for AED and RNN-T models, respectively.

For streaming encoders, we used the unidirectional LSTM (UniLSTM) and causal Conformer (UniConformer) [14] architectures, both of which follow four CNN layers with a 3×3 filter. A max-pooling layer with a stride of 2×2 was inserted at the 2nd and 4th CNN layer for LSTM and the 4th CNN layer for UniConformer. We used five layers of UniLSTM encoder with 1024 units [4]. We adopted Conformer (M) configuration in [14] while reducing the number of blocks from 16 to 12. Moreover, we replaced batch normalization [23] in each convolution module with layer normalization [24].[2] We used Shaw's style relative positional encoding (RPE) [26] and clipped the relative distance to 10 in each block. For streaming, we adopted causal depthwise separable convolution with a kernel size of 7 and a causal self-attention mask. Therefore, the lookahead latency was introduced in the frontend CNN layers only. Furthermore, we used hierarchical downsampling [27], where the max-pooling with a stride of 2 was performed in the last frontend CNN layer, 4th, and 8th Conformer blocks, respectively. This resulted in the downsampling factor of 8 for the UniConformer in total while 4 for the LSTM. The total lookahead latency in the UniLSTM and UniConformer encoders were 60ms and 40ms, respectively. The decoder consisted of a single layer of LSTM with 1024 units for MoChA, with a chunk size $w = 4$. For RNN-T, we used a two-layer LSTM prediction network with 1024 units and a joint network with 512 units.

The Adam algorithm [28] was used for optimization. We set λ_{ctc} to 0.3 in all models, including RNN-T.[3] We set $(\lambda_{\mathrm{latency}}, \lambda_{\mathrm{qua}})$ to (1.0, 0.0) for CTC-ST, otherwise (0.0, 2.0). During inference, we use the beam width b of 10 with a four-layer LSTM language model having 1024 units. We used length normalization [29] and softmax smoothing with a temperature of 0.7 [30]. Our implementation is publicly available.[4]

[2] We also tried group normalization [25], but layer normalization was the best in our implementations.

[3] RNN-T is optimized via $\mathcal{L}_{\mathrm{rnnt}}^{\mathrm{total}} = (1-\lambda_{\mathrm{ctc}})\mathcal{L}_{\mathrm{rnnt}} + \lambda_{\mathrm{ctc}}\mathcal{L}_{\mathrm{ctc}}$.

[4] `https://github.com/hirofumi0810/neural_sp`.

Table 2: *Results on TEDLIUM2. Token emission latency (TEL) is calculated on the test set.* ♠ *SpecAugment was NOT used.*

Model	WER [%] (↓)		TEL [ms] (↓)		
	dev	test	50th	90th	95th
BiLSTM - Global AED	8.1	7.5	–	–	–
UniLSTM - RNN-T ($b = 5$)	11.0	11.0	–	–	–
UniLSTM - RNN-T	10.7	10.7	–	–	–
UniLSTM - MoChA♠	16.3	14.9	280	680	1120
+ StableEmit♠	**13.9**	**12.7**	**240**	**440**	**640**
+ DeCoT-CTC ($\delta = 12$)	**12.4**	**10.7**	**240**	**400**	**480**
+ DeCoT-CTC ($\delta = 16$)	12.5	11.1	280	440	520
+ DeCoT ($\delta = 16$)	12.4	**10.7**	280	440	560
+ StableEmit	**12.0**	**10.7**	**240**	**360**	**440**
+ CTC-ST	13.5	11.3	200	360	480
+ StableEmit	**11.7**	**10.9**	200	**320**	**440**
Conformer - Global AED	7.3	7.0	–	–	–
UniConformer - CTC	12.1	11.2	240	320	400
UniConformer - RNN-T	8.5	8.2	–	–	–
UniConformer - MoChA	9.6	8.8	320	480	640
+ StableEmit	**9.1**	**8.4**	**240**	480	**480**
+ DeCoT-CTC ($\delta = 8$)	**8.8**	**8.4**	**240**	**400**	**480**
+ DeCoT-CTC ($\delta = 12$)	**8.5**	**8.4**	320	480	**480**
+ DeCoT ($\delta = 8$)	8.5	8.5	**240**	400	480
+ StableEmit	8.7	8.7	**240**	**320**	**400**
+ DeCoT ($\delta = 12$)	8.7	8.5	320	480	560
+ StableEmit	**8.6**	**8.2**	**240**	**400**	**480**
+ CTC-ST	**8.9**	8.8	**240**	**320**	**400**
+ StableEmit	9.3	**8.5**	**240**	**320**	**400**

4.2. Main results

We summarized the results in Table 2. We used forced alignment results to evaluate the token emission latency (TEL) [4] from the ground-truth acoustic boundary. We report median, 90th, and 95th percentile latency calculated on the *test* set.

4.2.1. UniLSTM encoder models

We first confirmed significant improvement by StableEmit over the baseline UniLSTM MoChA when SpecAugment was not used. We activated SpecAugment with alignment regularization methods after pre-training the baseline model to stabilize the training [7, 10]. While both CTC-ST and DeCoT were effective, the combination with StableEmit brought further improvements with an additional TEL reduction. We confirmed that DeCoT-CTC was also effective with a smaller δ than that in DeCoT because CTC also had a delay. Specifically, applying StableEmit to CTC-ST showed relative WER improvements of 13.3% (*dev*) and 3.5% (*test*), and absolute TEL reductions of 40ms and 80ms in the 90th and 95th percentile, respectively. Compared to the baseline model, the TEL was reduced by 80ms, 160ms, and 240ms in the 50th, 90th, and 95th percentile, respectively. We also observed that StableEmit was effective in reducing the tail of TEL distributions for DeCoT.

4.2.2. UniConformer encoder models

Next, we applied StableEmit to the UniConformer models. Unlike the UniLSTM models, we activated SpecAugment from scratch and activated $\mathcal{L}_{\mathrm{latency}}$ and $\mathcal{L}_{\mathrm{qua}}$ after running for 10 epochs, without pre-training. Firstly, we observed a significant improvement from the UniLSTM models with the Conformer encoder. StableEmit was also effective in this case, leading to relative WER improvements of 5.2% (*dev*) and 4.5% (*test*) and absolute reduction of the median TEL by 80ms. Leveraging alignment information further reduced the TEL and WER in some cases. In this regard, we observed that alignment path restriction (DeCoT and DeCoT-CTC) was more effective than

Table 3: *Effects of boundary threshold and loss weight on TEDLIUM2. The encoder is UniConformer.*

λ_{se}	τ	WER [%] (↓)		sub / ins / del	TEL [ms] (↓)	
		dev	test	dev	50th	90th
0.0	0.5	9.58	8.76	4.88 / 1.55 / 3.15	320	480
	0.4	**9.50**	8.48	4.96 / 1.77 / 2.77	320	480
	0.3	9.64	8.39	4.95 / 2.16 / 2.52	320	480
	0.2	10.11	**8.32**	5.00 / 2.79 / 2.31	320	480
	0.1	11.70	8.70	5.00 / 4.78 / 1.91	**240**	**400**
0.0*	0.5	9.80	9.00	4.92 / 1.45 / 3.43	320	560
0.1	0.5	9.13	8.45	4.94 / 1.48 / 2.71	240	480
	0.4	9.10	8.47	4.93 / 1.57 / 2.60	240	**400**
	0.3	**9.06**	8.39	5.06 / 1.61 / 2.40	240	**400**
	0.2	9.12	**8.29**	5.16 / 1.77 / 2.19	240	**400**
	0.1	9.63	8.45	5.26 / 2.28 / 2.15	240	**400**
0.1*	0.5	9.13	8.55	4.97 / 1.42 / 2.74	240	480
0.05	0.5	9.15	8.56	4.92 / 1.51 / 2.73	320	480
0.15		9.38	8.45	4.90 / 1.73 / 2.76	**240**	**400**
0.2		10.01	8.99	4.86 / 2.47 / 2.67	**240**	**400**

*λ_{se} was set to 0.1 at test time.

Table 4: *Ablation study on TEDLIUM2*

Model	WER [%] (↓)	
	dev	test
UniConformer - MoChA + StableEmit	**9.1**	**8.4**
w/o quantity reg. ($\lambda_{\text{qua}} = 0$ in Eq. (3))	9.7	9.9
w/o StableEmit ($\lambda_{\text{se}} = 0$ in Eq. (5))	9.6	8.8
w/o hierarchical downsampling	10.3	10.0
w/o StableEmit	11.1	10.4
w/o CTC loss ($\lambda_{\text{ctc}} = 0$ in Eq. (3))	9.2	8.8
LayerNorm → BatchNorm	31.3	23.4
UniConformer - RNN-T	**8.5**	**8.2**
w/o hierarchical downsampling	8.7	8.4
w/o CTC loss ($\lambda_{\text{ctc}} = 0$)	8.7	8.6
LayerNorm → BatchNorm	8.8	**8.2**

CTC-ST in terms of WER while CTC-ST reached the same level of TEL as the pure CTC model without external alignment. Although the gains became smaller than those of the UniLSTM models, we attributed it to the larger downsampling factor, which improved the baseline performance significantly, as shown in Section 4.4.

4.3. Analysis

Next, we study the effect of a discount factor λ_{se} (for training) and a threshold τ for selection probabilities (for testing). We used the UniConformer encoder for this purpose. The results in Table 3 showed that reducing τ from 0.5 slightly reduced WER regardless of the use of StableEmit. However, the model trained with StableEmit consistently outperformed the baseline in both WER and TEL and was robust to a small τ. Furthermore, we also set λ_{se} to 0.1 during testing, but this did not improve WER nor TEL. This verifies that the confidence of $p_{i,j}$ at test time is increased by StableEmit. For simplicity, we keep $\tau = 0.5$ in all other experiments.

We also investigate the effect of different values of λ_{se}. Increasing λ_{se} reduced the 90th percentile TEL by 80ms while at the cost of WER. Regarding error types, StableEmit successfully reduced the deletion errors while a strong constraint by a small τ or a large λ_{se} resulted in increasing the insertion errors.

4.4. Ablation study

We also conduct the ablation study in Table 4. We observed that the quantity regularization was essential for StableEmit. The hierarchical downsampling and layer normalization were also key ingredients when the UniConfomer encoder was used. Moreover, we found that the hierarchical downsampling, the auxiliary CTC loss, and layer normalization were also beneficial for RNN-T.

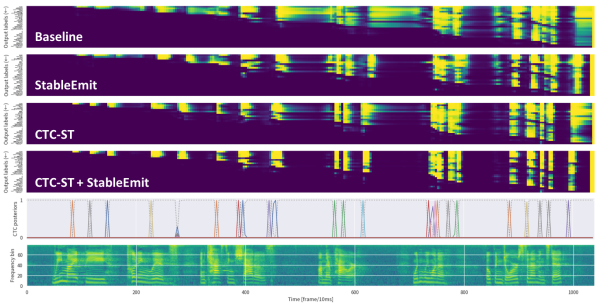

Figure 1: *Visualization of selection probabilities and CTC outputs from UniLSTM MoChA and the input spectrogram. Yellow regions represent high probabilities.*

Table 5: *Results on Librispeech. TEL was averaged over the test-clean and test-other sets.*

Model	WER [%] (↓)				TEL [ms] (↓)	
	dev		test		50th	90th
	clean	other	clean	other		
UniConformer - RNN-T	2.8	8.2	3.1	8.5	–	–
UniConformer - MoChA	2.6	7.6	2.8	7.9	400	560
+ StableEmit	2.9	**7.5**	3.0	8.2	**320**	**480**

4.5. Visualization of selection probabilities

We visualize the selection probabilities $p_{i,j}$ in the UniLSTM MoChA models in Figure 1. The yellow regions represent high probabilities close to 1. We used the whole encoder outputs for visualization. We also plotted the CTC probabilities obtained from the model trained with CTC-ST and StableEmit. We can see that the baseline MoChA had blurred probabilities, and StableEmit sharpened the distribution, indicating the confidence to emit a token was increased. CTC-ST shifted some token boundaries to the left and sharpened the probability distribution [4]. Moreover, its combination with StableEmit further made them clearer, showing a complementary effect. The delay became almost the same as that of CTC.

4.6. Results on Librispeech

Finally, we present results on Librispeech (960h) [31] in Table 5. We used the same architecture as previous experiments. We used $\lambda_{\text{qua}} = 0.1$ for the baseline and (λ_{qua}, λ_{se}) = (0.2, 0.2) for StableEmit. We observed TEL reductions of 80ms in the 50th and 90th percentile at the cost of a slight WER degradation. This is because MoChA did not suffer from the deletion errors on this corpus. However, it outperformed a strong RNN-T model having the same encoder architecture.

5. Conclusions

In this work, we have proposed StableEmit, a simple alignment-free solution to reduce the emission latency of MoChA for streaming ASR applications. We encouraged the model to learn high selection probabilities in the token boundary detection by discounting them during training as a regularization. We experimentally demonstrated that StableEmit reduced not only the emission latency but also the deletion errors. Combining it with the alignment regularization further reduced the latency.

6. References

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of ICML*, 2006, pp. 369–376.

[2] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[3] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *Proceedings of ICLR*, 2018.

[4] H. Inaguma and T. Kawahara, "Alignment knowledge distillation for online streaming attention-based speech recognition," *arXiv preprint arXiv:2103.00422*, 2021.

[5] C. Kim, D. Gowda, D. Lee, J. Kim, A. Kumar, S. Kim, A. Garg, and C. Han, "A review of on-device fully neural end-to-end automatic speech recognition algorithms," *arXiv preprint arXiv:2012.07974*, 2020.

[6] H. Sak, F. de Chaumont Quitry, T. Sainath, K. Rao *et al.*, "Acoustic modelling with CD-CTC-sMBR LSTM RNNs," in *Proceedings of ASRU*. IEEE, 2015, pp. 604–609.

[7] H. Inaguma, Y. Gaur, L. Lu, J. Li, , and Y. Gong, "Minimum latency training strategies for streaming sequence-to-sequence ASR," in *Proceedings of ICASSP*. IEEE, 2020, pp. 6064–6068.

[8] J. Mahadeokar, Y. Shangguan, D. Le, G. Keren, H. Su, T. Le, C.-F. Yeh, C. Fuegen, and M. L. Seltzer, "Alignment restricted streaming recurrent neural network transducer," in *Proceedings of SLT*. IEEE, 2021, pp. 52–59.

[9] H. Hu, R. Zhao, J. Li, L. Lu, and Y. Gong, "Exploring pre-training with alignments for RNN transducer based end-to-end speech recognition," in *Proceedings of ICASSP*. IEEE, 2020, pp. 7079–7083.

[10] H. Inaguma, M. Mimura, and T. Kawahara, "CTC-synchronous training for monotonic attention model," in *Proceedings of Interspeech*, 2020, pp. 571–575.

[11] J. Yu, C.-C. Chiu, B. Li, S.-y. Chang, T. N. Sainath, Y. He, A. Narayanan, W. Han, A. Gulati, Y. Wu *et al.*, "FastEmit: Low-latency streaming asr with sequence-level emission regularization," in *Proceedings of ICASSP*. IEEE, 2021, pp. 6004–6008.

[12] J. Yu, W. Han, A. Gulati, C.-C. Chiu, B. Li, T. N. Sainath, Y. Wu, and R. Pang, "Dual-mode ASR: Unify and improve streaming ASR with full-context modeling," in *Proceedings of ICLR*, 2021.

[13] J. Kim, H. Lu, A. Tripathi, Q. Zhang, and H. Sak, "Reducing streaming ASR model delay with self alignment," *arXiv preprint arXiv:2105.05005*, 2021.

[14] 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 *Proceedings of Interspeech*, 2020, pp. 5036–5040.

[15] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," in *Proceedings of ICML*, 2017, pp. 2837–2846.

[16] H. Miao, G. Cheng, P. Zhang, T. Li, and Y. Yan, "Online hybrid CTC/attention architecture for end-to-end speech recognition," in *Proceedings of Interspeech*, 2019, pp. 2623–2627.

[17] ——, "Online hybrid CTC/attention end-to-end automatic speech recognition architecture," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1452–1465, 2020.

[18] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: An automatic speech recognition dedicated corpus," in *Proceedings of LREC*, 2012, pp. 125–129.

[19] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proceedings of ASRU*. IEEE, 2011.

[20] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proceedings of Interspeech*, 2015, pp. 3586–3589.

[21] 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 *Proceedings of Interspeech*, 2019, pp. 2613–2617.

[22] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of ACL*, 2016, pp. 1715–1725.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of ICML*, 2015, pp. 448–456.

[24] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[25] Y. Wu and K. He, "Group normalization," in *Proceedings of ECCV*, 2018, pp. 3–19.

[26] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of NAACL-HLT*, 2018, pp. 464–468.

[27] 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 *Proceedings of ICASSP*, 2019, pp. 5656–5660.

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] K. Murray and D. Chiang, "Correcting length bias in neural machine translation," in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Association for Computational Linguistics, 2018, pp. 212–223.

[30] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proceedings of Interspeech*, 2017, pp. 523–527.

[31] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proceedings of ICASSP*. IEEE, 2015, pp. 5206–5210.