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

Reducing Streaming ASR Model Delay with Self Alignment

Jaeyoung Kim, Han Lu, Anshuman Tripathi, Qian Zhang, Hasim Sak

Google Inc., USA

{jaeykim, luha, anshumant, zhaqian, hasim}@google.com

Abstract

Reducing prediction delay for streaming end-to-end ASR models with minimal performance regression is a challenging problem. Constrained alignment is a well-known existing approach that penalizes predicted word boundaries using external low-latency acoustic models. On the contrary, recently proposed FastEmit is a sequence-level delay regularization scheme encouraging vocabulary tokens over blanks without any reference alignments. Although all these schemes are successful in reducing delay, ASR word error rate (WER) often severely degrades after applying these delay constraining schemes. In this paper, we propose a novel delay constraining method, named self alignment. Self alignment does not require external alignment models. Instead, it utilizes Viterbi forced-alignments from the trained model to find the lower latency alignment direction. From LibriSpeech evaluation, self alignment outperformed existing schemes: 25% and 56% less delay compared to FastEmit and constrained alignment at the similar word error rate. For Voice Search evaluation, 12% and 25% delay reductions were achieved compared to FastEmit and constrained alignment with more than 2% WER improvements.

Index Terms: Transformer, RNN-T, sequence-to-sequence, encoder-decoder, end-to-end, speech recognition

1. Introduction

End-to-end ASR models [1, 2, 3] have successfully expanded their presence by showing performance as competitive as conventional hybrid models [4]. In particular, recurrent neural network transducer (RNN-T) architecture [2] gained the most popularity among them due to its natural streaming capability as well as superior performance. Transformer-Transducer [3, 5, 6] further improved RNN-T architecture by replacing LSTMs with Transformer layers [7]. However, streaming end-to-end models which optimize sequence likelihoods without any delay constraints suffer from high delay between the audio input and the predicted text because models learn to improve their prediction by using more future context.

Recently, there have been several approaches to reduce prediction delay. The constrained alignment [8, 9, 10] penalizes word boundaries based on audio alignment information from an external alignment model by masking out alignment paths exceeding the predetermined threshold delay. Inaguma *et al.* [10] proposed frame-wise CE regularization by constraining encoder output using a conventional hybrid model. FastEmit [11], unlike teacher-assisted training schemes, does not need external alignments. It is a sequence-level delay regularization scheme which encourages vocabulary tokens over blank one across the entire alignment paths in the RNN-T decoding graph.

Although previous delay improving schemes can reduce latency of streaming end-to-end models, there are several issues on them. Teacher-assisted schemes such as constrained alignment heavily depend on the performance of external models. Since performance regression always happens when the model latency decreases, high-precision external alignment models would be needed to minimize WER degradation, which can further complicate model training steps. On the contrary, FastEmit blindly reduces delay by choosing the most efficient direction in the RNN-T decoding graph. However, its direction might not be optimal for all audio input due to lack of alignment information, which can degrade delay-WER trade-offs.

In this paper, we propose a novel delay constraining method, self alignment. Self alignment does not require external alignment models similar to FastEmit. However, the main difference is that it does not blindly optimize delay but utilizes Viterbi forced-alignments from the trained model to find the better low latency direction. For example, self alignment always finds the path one frame left to the Viterbi forced alignment and optimize it with the main objective. Therefore, self alignment keeps pushing the main alignment path to its left direction for each training step.

The proposed self alignment scheme has advantages over existing schemes. First, training complexity for self alignment is much lower than teacher-assisted schemes since it does not need external alignment models. Second, self alignment minimally affects ASR training by only constraining the most probable alignment path. On the contrary, other schemes affect many alignment paths by masking out them or changing weights on their label transition probabilities. Since delay constraining regularization terms always conflict with the main ASR loss, minimal intervention on the main loss would be important to optimizing delay and performance trade-offs. Self alignment only regularizes single path by pushing it to its left direction. Third, self alignment provides better low latency direction than FastEmit. From LibriSpeech evaluation, self alignment provides 25% and 56% less delay compared to FastEmit and constrained alignment when WER is fixed at around 4%. For large data evaluation, self alignment still provides better delay-WER trade-offs on Voice Search data: 12% and 25% delay reductions compared to FastEmit and constrained alignment, meanwhile maintaining better WER performance.

2. Transformer Transducers

Transformer-Transducer (T-T) was introduced in [3, 5, 6] by replacing RNN-based audio and label encoders with Transformer models in RNN-T architecture [2]. Figure 1 depicts T-T architecture. For each time step, a T-T model provides alignment distribution $P(\mathbf{z}|\mathbf{x})$ based on the past labels and audio signals which can be factorized as follows:

$$\Pr(\mathbf{z}|\mathbf{x}) = \prod_i \Pr(z_i|\mathbf{x}, t_i, \mathbf{y}_{1:u^i}) \tag{1}$$

where $\mathbf{x}$ is audio input, $\mathbf{y}$ is a ground-truth label sequence, $\mathbf{z}$ is an alignment belonging to $\mathbf{y}$, $\mathbf{y}_{1:u^i}$ is a partial label sequence processed from an alignment $\mathbf{z}_{1:(i-1)}$ by filtering blank symbols. Then the log conditional probability of $\mathbf{y}$ given audio in-

http://dx.doi.org/10.21437/Interspeech.2021-322

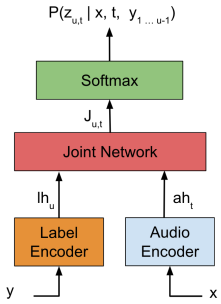

Figure 1: *Transformer Transducer architecture.*

put $\mathbf{x}$ is to sum all the alignment distributions corresponding to $\mathbf{y}$ as follows:

$$\log \Pr(\mathbf{y}|\mathbf{x}) = \log \sum_{K(\mathbf{z})=\mathbf{y}} \Pr(\mathbf{z}|\mathbf{x}) \tag{2}$$

where the mapping K removes blank symbols in $\mathbf{z}$.

The log total alignment probability in Eq. 2 is a target loss function which can be efficiently computed using forward-backward algorithm as follows:

$$\Pr(\mathbf{y}|\mathbf{x}) = \alpha(T, U) \tag{3}$$

$$\begin{aligned}\alpha(t,u) = \alpha(t-1,u-1)\Pr(\phi|t-1,u)+ \\ \alpha(t,u-1)\Pr(y_u|t,u-1)\end{aligned} \tag{4}$$

where $\Pr(\phi|t-1,u)$ and $\Pr(y_u|t,u-1)$ are blank and label probabilities and T and U are audio and label sequence lengths.

3. Alignment Delay

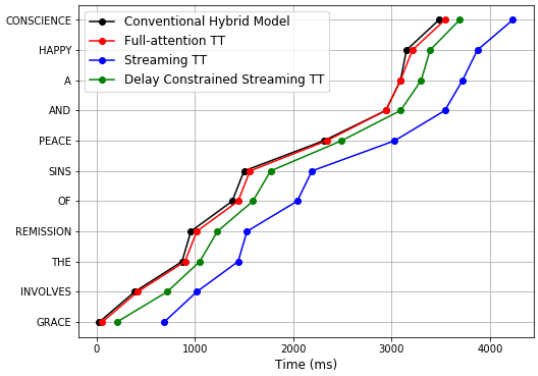

Figure 2: *Evaluation of alignment delay*

In this section, we compare alignment delay between conventional acoustic model with cross entropy training based on Viterbi forced-alignments [4] and T-T models with the total alignment probability as in Eq. 2. Alignment delay is delay between input audio frames and streamed decoded labels. Since label alignments can be gradually enhanced by iteratively training aligning models with realigned labels, a conventional model can learn accurate alignments after multiple iterations. In this section, alignments from a conventional model are assumed to be correct and used as ground-truth.

Table 1: *Delay and WER comparison between streaming models with and without applying constrained alignment: Both average and RMS delays are evaluated on 100 Libri speech test clean samples.*

Streaming T-T	Mean Delay	RMS Delay	TestClean	TestOther
baseline	610ms	615ms	**3.4%**	**9.5%**
delay const.	**172ms**	**175ms**	4.4%	11.9%

Figure 2 shows Viterbi forced-alignments for different T-T models. The x-axis is audio frame time in milliseconds and the y-axis is a label sequence. A full-attention T-T model can access future frames when it computes self-attentions at Transformer layers. Its alignment path in Figure 2 almost coincides with the one from the conventional model. However, streaming T-T whose self-attention only depends on past frames showed excessive delay, more than 600 ms compared to full-attention T-T. Since the total alignment probability in Eq. 2 includes all the label alignments without any delay constraint, a T-T model can arbitrarily increase its prediction delay in order to improve model accuracy by accessing to future context.

Basically, the alignment delay is a critical issue for streaming ASR models. There are several existing delay constraining schemes for end-to-end training. The green alignment path at Figure 2 came from one of them, constrained alignment described at Section 4.1. After applying constrained alignment, the overall delay is significantly reduced from 610 ms to 172 ms at Table 1. However, the model accuracy for the delay constrained model severely degraded due to the reduced future context. The mean and RMS delay definitions are explained at Section 6.1. The delay constrained model reduced average delay around 440 ms but test clean WER degraded close to 30%. Therefore, in order to properly evaluate alignment schemes, both delay and WER should be considered together.

4. Related Works

4.1. Constrained Alignment

Constrained alignment training was first introduced for CTC models in [8] and later extended to RNN-T and monotonic chunkwise attention (MoChA) models [9, 10]. It constrains the total alignment probability loss in Eq. 2 by completely masking alignment paths with delay exceeding predetermined threshold. The delay is measured from the ground-truth alignment which can be supplied from external alignment models. There is also an extension to soft constrained alignment [9], where paths outside of the threshold delay are scaled by weights inversely proportional to the distance from the ground-truth alignments.

Figure 3 shows an example of a decoding graph for a label sequence 'I like it'. The threshold delay is enforced only to a word boundary token, a space symbol. The red alignment path at Figure 3 is a reference alignment and the purple alignment is the rightmost allowed path when word boundary threshold is set to be 2. The constrained alignment for word boundary tokens can be formulated in the forward algorithm as follows:

$$\begin{aligned}\alpha(t,u) =& \alpha(t-1,u-1)\Pr(\phi|t-1,u)+ \\ & \alpha(t,u-1)\Pr(y_u|t,u-1)\mathbb{1}(t < T_u + \sigma)\end{aligned} \tag{5}$$

where T_u is ground-truth alignment time for u^{th} token, σ is

threshold and $\mathbb{1}$ is an indicator function. Eq. 5 is only applied when u^{th} label is a word boundary token. Otherwise, the original forward algorithm at Eq. 4 is used.

4.2. FastEmit

FastEmit is a newly proposed delay-constrained scheme [11]. It does not require external alignments, which is a main advantage over constrained alignment or encoder distillation. It directly manipulates forward-backward sequence probabilities by boosting the probability of the next vocabulary token and discouraging the blank transition at all (u, t) positions. For example, FastEmit pushes alignment paths to the upper left direction in Figure 3. FastEmit training can be implemented by simple gradient boosting technique as in Eq. 11 and Eq. 12 from [11].

5. Proposed Scheme

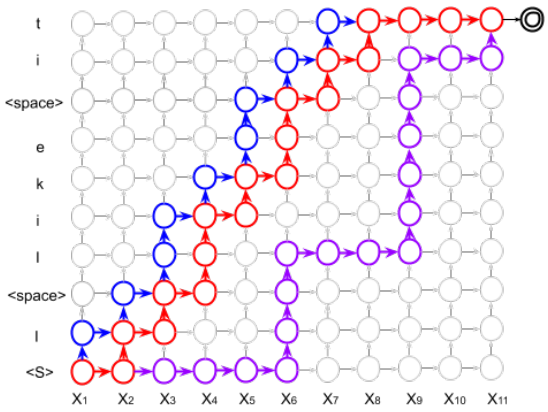

Figure 3: *T-T decoding graph for a label sequence, "I like it": The red alignment path is a reference alignment derived from either an external alignment model or delay-constrained model itself.*

We propose a new delay improving scheme, named self alignment. The self alignment scheme does not need external alignment models. Instead, Viterbi forced-alignments from the trained model itself are used to find the delay improving direction, which is one frame left to the forced-alignment path. For example, in Figure 3, the red alignment path is the forced-alignment from the trained model and the blue alignment is one frame left to it. For each training batch, the self alignment scheme is to encourage the left alignment paths, which constantly pushes the model's forced-alignments to the left direction. The training loss can be formulated as follows:

$$\mathcal{L}_{\text{total}} = -\log \Pr(\mathbf{y}|\mathbf{x}) - \lambda \sum_{u} \log \Pr\left(y_u | t_u, u\right) \qquad (6)$$

where λ is a weighting factor for the left-alignment likelihoods, t_u is a frame index for the left alignment at the u^{th} token.

The proposed self alignment scheme has advantages over the existing schemes seen at Section 4. First, like FastEmit, it does not need external alignment models. Constrained alignment needs external teacher models for reference alignments and, therefore, their performance heavily depends on the quality of teacher models. Second, self alignment can push alignments for the training model to the better direction than FastEmit. FastEmit provides good delay-WER trade-off without increasing memory and computational complexity. However, FastEmit blindly pushes alignments to the left upper direction in the decoding graph, which is clearly not optimal direction for all the audio input. Self alignment utilizes its own forced-alignments to guess the next alignment direction.

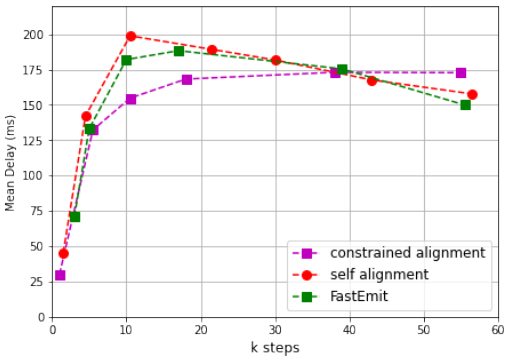

Figure 4: *Comparison of delay training curves*

Figure 4 compared mean delay for each training steps. For constrained alignment, delay regularization is strictly applied to all training steps because mean delay is monotonically increasing. Delay regularization usually conflicts with main loss and the strict regularization for all training steps would make training hard for constrained alignment. On the other hand, FastEmit and self alignment showed delay constraint is not dominant for initial training steps. Mean delays initially increase for minimizing main RNN-T loss but when main loss is close to convergence, delay starts to decrease where delay regularization is getting dominant. This trend especially makes sense for self alignment. Initially, training is focused on the main loss but when Viterbi forced alignments are getting to make sense, training is shifting to reduce the delay of the Viterbi paths.

6. Experiments and Results

6.1. Setup

We evaluate delay improving schemes on two separate datasets: LibriSpeech [12] and Google's Voice Search. LibriSpeech corpus is a read speech data based on audio books and consists of 1000 hours of training and test sets. For larger data evaluation, we use Google Voice Search corpus with more than 30k hours. The test set contains 14k Voice Search utterances with length less than 5.5 seconds. The training and test sets are all anonymized and hand-transcribed. The input audio is processed as 128 dimension logmel energy features and stacked with 4 frames. Before fed as acoustic features, stacked logmels are subsampled by a factor of 3, resulting in 30ms features. For robust training, specaugment [13] is applied to acoustic features.

We use two delay metrics to compare different schemes: 1) mean alignment delay and 2) root mean square (RMS) delay. The mean alignment delay is defined as mean word time difference between ground-truth and predicted alignments:

$$D_{\text{mean}} = \frac{1}{\sum_{k=1}^{N} |\mathbf{y}_k|} \sum_{k=1}^{N} \sum_{i=1}^{|\mathbf{y}_k|} (\hat{t}_i^k - t_i^k) \qquad (7)$$

where t_i^k is i^{th} word time from a reference model, $\hat{t}_i^k$ is i^{th} predicted word time, $|\mathbf{y}_k|$ is k^{th} utterance word length and N is the number of utterances. One issue for the mean alignment delay is that it cannot reflect delay variance in the metric. RMS delay

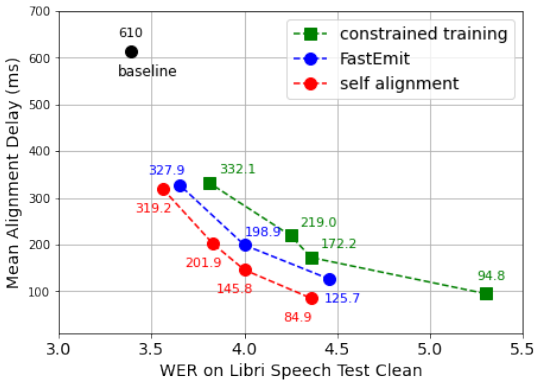

Figure 5: *Delay-WER trade-off comparison on Libri speech test clean*

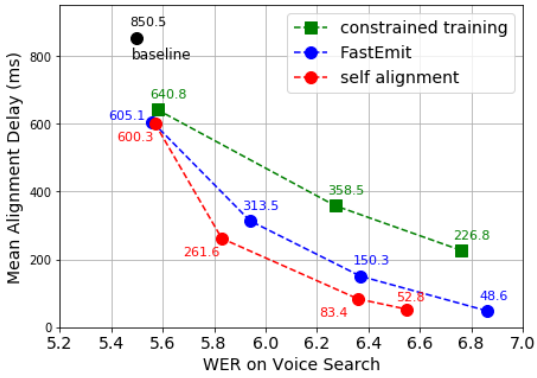

Figure 6: *Delay-WER trade-off comparison on Voice Search.*

Table 2: *Delay and WER evaluation on LibriSpeech at around 4% test clean WER: Both mean and RMS delays were evaluated on 100 LibriSpeech test clean samples.*

Model	Mean Delay	RMS Delay	TestClean	TestOther
A Streaming T-T	610ms	615ms	3.4%	9.5%
A + ConstAlign	328ms	330ms	4.0%	11.1%
A + FastEmit	195ms	215ms	4.0%	**10.4%**
A + SelfAlign	**145ms**	**164ms**	4.0%	10.7%

Table 3: *Delay and WER evaluation on VS: Both mean and RMS delays were evaluated on 100 LibriSpeech test clean samples.*

Model	Mean Delay	RMS Delay	VS WER
A Streaming T-T	856ms	858ms	5.5%
A + ConstAlign	358ms	360ms	6.3%
A + FastEmit	307ms	331ms	5.9%
A + SelfAlign	**269ms**	**278ms**	**5.8%**

can measure delay variance and defined as follows:

$$D_{\text{rms}} = \sqrt{\frac{1}{\sum_{k=1}^{N} |\mathbf{y}_k|} \sum_{k=1}^{N} \sum_{i=1}^{|\mathbf{y}_k|} (\hat{t}_i^k - t_i^k)^2} \tag{8}$$

6.2. Main Results on Libri Speech

The streaming T-T audio encoder for LibriSpeech consists of 15 Transformer layers with 100 left context frames for each layer. The streaming T-T model is trained with an output delay of 4 frames and has a character-based label encoder which has 2 Transformer layers with 20 left context symbols.

Figure 5 compares delay-WER trade-offs on LibriSpeech test clean data. Delay-WER trade-offs can be obtained by sweeping delay tuning hyper-parameters. For example, constrained alignment can adjust the model delay by tuning σ at Eq. 5. For other schemes, λ serves as a tuning hyper-parameter. In Figure 5, self alignment outperformed all other schemes from low to high latency region. For high latency region, FastEmit and self alignment showed similar delay-WER trade-offs. Their gap gradually increases as the model moves into the low latency region. Constrained alignment showed poor delay-WER trade-offs, which is a surprising result considering it utilized external alignment models. Full-attention T-T was used as an alignment model which can have higher variance compared to a conventional model because a full-attention T-T was not directly trained with alignment labels. The second reason is that delay constraint is imposed for all training steps for constrained alignment which could make training harder than other schemes. Although not used in this evaluation, soft constrained alignment [9] could make training more easier because it does not completely mask out outside of the delay threshold.

Table 2 compared mean and RMS delay on LibriSpeech at around 4% WER. Self alignment showed 25% and 56% less mean delay compared to FastEmit and constrained alignment, respectively.

6.3. Evaluation on Voice Search Corpus

In this section, we evaluate self alignment on the larger audio corpus, Voice Search (VS) data. The audio encoder for Voice Search has 18 Transformer layers with 32 left context frames. The label encoder has the same architecture as for Libri Speech. Figure 6 compares Delay-WER curves for constrained training, FastEmit and self alignment. Similar to the LibriSpeech result, self alignment outperformed other schemes for all latency region. For high latency region, FastEmit and self alignment showed less than 0.1% WER loss with more than 250ms delay reduction. As moving into the lower latency region, their WER gap gradually increases. Constrained training still showed poor delay-WER trade-off compared to self alignment and FastEmit. Table 3 compared different schemes with mean delays around 300ms. Self alignment showed 12% and 2% relative improvements on delay and WER over FastEmit. Compared to the constrained alignment, self alignment presented 25% and 7% relative gains on delay and WER, respectively.

7. Conclusions

In this paper, we proposed a novel delay constraining method, self alignment. Self alignment uses Viterbi forced-alignments from the trained model to find the lower latency alignment direction. Since it does not use external alignment models, training steps are much simpler than the constrained alignment scheme. Moreover, Viterbi forced-alignments can provide the better delay improving direction than FastEmit. The experimental result showed that the proposed self alignment significantly outperformed existing approaches on delay-WER trade-offs for both LibriSpeech and Voice Search datasets.

8. 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*, 2006, pp. 369–376.

[2] A. Graves, "Sequence transduction with recurrent neural networks," in *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[3] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," 2020.

[4] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.

[5] C.-F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le, M. Jain, K. Schubert, C. Fuegen, and M. L. Seltzer, "Transformer-transducer: End-to-end speech recognition with self-attention," 2019.

[6] A. Tripathi, J. Kim, Q. Zhang, H. Lu, and H. Sak, "Transformer transducer: One model unifying streaming and non-streaming speech recognition," *arXiv preprint arXiv:2010.03192*, 2020.

[7] 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.

[8] H. Sak, F. de Chaumont Quitry, T. Sainath, K. Rao *et al.*, "Acoustic modelling with cd-ctc-smbr lstm rnns," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 604–609.

[9] T. N. Sainath, R. Pang, D. Rybach, B. Garcıa, and T. Strohman, "Emitting word timings with end-to-end models," *Proc. Interspeech 2020*, pp. 3615–3619, 2020.

[10] H. Inaguma, Y. Gaur, L. Lu, J. Li, and Y. Gong, "Minimum latency training strategies for streaming sequence-to-sequence asr," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6064–6068.

[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," *arXiv preprint arXiv:2010.11148*, 2020.

[12] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. ICASSP*. IEEE, 2015, pp. 5206–5210.

[13] 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," *arXiv preprint arXiv:1904.08779*, 2019.