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

Advanced Long-context End-to-end Speech Recognition Using Context-expanded Transformers

Takaaki Hori, Niko Moritz, Chiori Hori, Jonathan Le Roux

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
{thori, moritz, chori, leroux}@merl.com

Abstract

This paper addresses end-to-end automatic speech recognition (ASR) for long audio recordings such as lecture and conversational speeches. Most end-to-end ASR models are designed to recognize independent utterances, but contextual information (e.g., speaker or topic) over multiple utterances is known to be useful for ASR. In our prior work, we proposed a context-expanded Transformer that accepts multiple consecutive utterances at the same time and predicts an output sequence for the last utterance, achieving 5-15% relative error reduction from utterance-based baselines in lecture and conversational ASR benchmarks. Although the results have shown remarkable performance gain, there is still potential to further improve the model architecture and the decoding process. In this paper, we extend our prior work by (1) introducing the Conformer architecture to further improve the accuracy, (2) accelerating the decoding process with a novel activation recycling technique, and (3) enabling streaming decoding with triggered attention. We demonstrate that the extended Transformer provides state-of-the-art end-to-end ASR performance, obtaining a 17.3% character error rate for the HKUST dataset and 12.0%/6.3% word error rates for the Switchboard-300 Eval2000 CallHome/Switchboard test sets. The new decoding method reduces decoding time by more than 50% and further enables streaming ASR with limited accuracy degradation.

Index Terms: end-to-end speech recognition, transformer, conformer, long context ASR

1. Introduction

Recent studies have produced different types of end-to-end models applicable for automatic speech recognition (ASR), such as connectionist temporal classification (CTC) [1], attention-based encoder decoder [2,3], RNN Transducer (RNN-T) [4], Transformer [5], and their combinations [6–9]. Specifically, Transformer has recently provided significant performance gain over RNN-based models in major sequence-to-sequence tasks including ASR [10, 11].

However, most ASR systems are designed to recognize independent utterances, despite the fact that contextual information over multiple utterances, such as information on the speaker or topic, is known to be useful for ASR. There are several approaches to incorporating contextual information in end-to-end ASR, such as i-vector approaches that utilize speaker context [12–15] and hierarchical RNN decoders that utilize discourse context [16, 17]. Besides, RNN-T and attention models have been applied to long-form ASR [18] , although the focus was on the scalability to long-form speeches in the inference phase.

In [19], we proposed a context-expanded Transformer, which extends the Transformer model to incorporate contextual information in training and decoding to improve the recognition accuracy for lecture and conversational speeches. The proposed method concatenates multiple adjacent utterances and trains a Transformer to recognize the last of these utterances. The previous utterances can thus be used to normalize or adapt the acoustic and linguistic features at every encoder/decoder layer for recognizing the last utterance. Moreover, we proposed to use speaker-dependent context, i.e., concatenating the utterances spoken by the same speaker. The proposed method achieved 5-15% relative error reduction from utterance-based baselines in lecture and conversational ASR benchmarks.

While our approach has shown substantial accuracy improvement in long-form ASR, some issues remain to be addressed:

1. it is unclear whether the approach is also effective for other Transformer-based models or not, in particular whether equivalent accuracy gains can be obtained for state-of-the-art architectures such as Conformers;
2. a high computational complexity is required in decoding due to the long speech input, as the model needs to process a long speech segment including multiple utterances to recognize the last utterance of the segment using self-attention/source-attention mechanisms, whose computational complexity increases quadratically with the segment length;
3. the method is not yet applicable for streaming ASR, which is indispensable for online applications.

In this paper, we address the above mentioned issues and investigate (1) introducing the Conformer architecture [20] to further improve accuracy, (2) accelerating the decoding process with a novel activation recycling technique, and (3) enabling streaming decoding with restricted self-attention and triggered attention [21–23]. We demonstrate the effectiveness of the extended Transformer and its decoding method using conversational ASR benchmarks on the HKUST [24] and Switchboard [25] corpora, achieving state-of-the-art accuracy and faster decoding compared with the original approach.

2. Context-expanded Transformer

Transformer [5] consists of encoder and decoder networks, which have deep feed-forward architectures including repeated blocks of self-attention and feed-forward layers with residual connections [26] and layer normalization [27]. The decoder network also features a source attention layer in each block to read the encoder's output.

Figure 1 illustrates the context-expanded Transformer proposed in our prior work [19]. The network architecture is basically the same as the original Transformer, but it accepts multiple utterances at once and predicts output tokens for the last utterance using previous utterances as contextual information.

http://dx.doi.org/10.21437/Interspeech.2021-1643

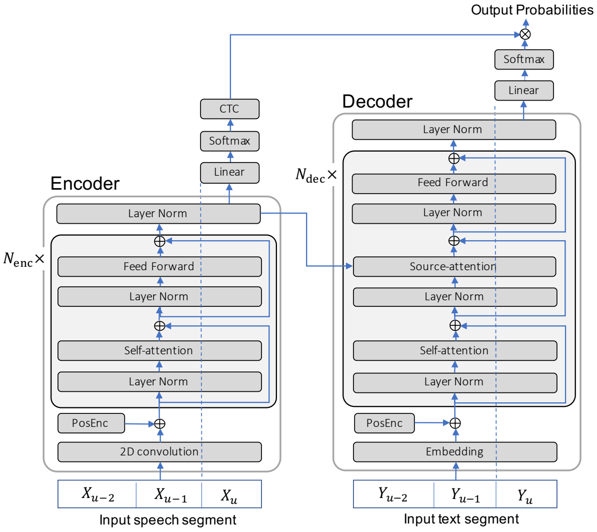

Figure 1: *Context-expanded Transformer [19]*

Given a feature sequence X_u for a u-th utterance and feature sequences $X_v, \dots, X_{u-1}$ for its previous utterances, where $1 \leq v < u$, we denote the input speech segment as $X_{v:u} = (X_v, \dots, X_{u-1}, X_u)$ and its corresponding output segment as $Y_{v:u} = (Y_v, \dots, Y_{u-1}, Y_u)$. The goal of ASR here is to find the most probable token sequence $\hat{Y}_u$ for the u-th utterance as

$$\begin{aligned}\hat{Y}_u &= \underset{Y_u \in \mathcal{V}^*}{\operatorname{argmax}}\, p(Y_u | Y_{v:u-1}, X_{v:u}) \\ &= \underset{Y_u = y_{u,1:L} \in \mathcal{V}^*}{\operatorname{argmax}} \prod_{i=1}^{L} p(y_{u,i} | Y_{v:u-1}, y_{u,1:i-1}, X_{v:u}), \quad (1)\end{aligned}$$

where $y_{u,1:L}$ denotes the token sequence $(y_{u,1}, \dots, y_{u,L})$ of Y_u, and $\mathcal{V}$ is the vocabulary.

The probability of $y_{u,i}$ in Eq. (1) is computed using the Transformer. The encoder first applies 2D convolution (Conv2D) and positional encoding (PosEnc) to all frames of $X_{v:u}$ and adds them to obtain the first hidden vector sequence

$$H^0_{v:u} = \mathrm{Conv2D}(X_{v:u}) + \mathrm{PosEnc}(X_{v:u}). \quad (2)$$

Then, it computes a hidden vector sequence in each encoder block, the sequence after the n-th block being obtained as

$$\bar{H}^{n-1}_{v:u} = \xi(H^{n-1}_{v:u}) \quad (3)$$

$$\tilde{H}^{n}_{v:u} = H^{n-1}_{v:u} + \mathrm{MHA}(\bar{H}^{n-1}_{v:u}, \bar{H}^{n-1}_{v:u}, \bar{H}^{n-1}_{v:u}) \quad (4)$$

$$H^{n}_{v:u} = \tilde{H}^{n}_{v:u} + \mathrm{FFN}(\xi(\tilde{H}^{n}_{v:u})), \quad (5)$$

where $\mathrm{MHA}(\cdot,\cdot,\cdot)$, $\mathrm{FFN}(\cdot)$, and $\xi(\cdot)$ represent multi-head attention, feed-forward network, and layer normalization, respectively. $\mathrm{MHA}()$ takes three arguments Q, K, and V, which are query, key, and value vector sequences [5]. For self-attention in the encoder, these arguments are equal to $\bar{H}^{n-1}_{v:u}$. The encoder states are obtained as the normalized output of the last block, i.e., $\bar{H}^{N_{\text{enc}}}_{v:u} = \xi(H^{N_{\text{enc}}}_{v:u})$, where N_{enc} denotes the number of encoder blocks.

The decoder accepts previous output sequence $(Y_{v:u-1}, y_{u,1:i-1})$ and the encoder states $\bar{H}^{N_{\text{enc}}}_{v:u}$, and estimates the probability distribution of $y_{u,i}$ in Eq. (1). For simplicity, we rewrite $(Y_{v:u-1}, y_{u,1:i-1})$ as $y'_{u,1:k-1}$, which represents all previous tokens up to index $k-1$ in the whole segment, where $|Y_{v:u-1}| < k \leq |Y_{v:u}|$ and $k = |Y_{v:u-1}| + i$, $|Y_*|$ denoting the number of tokens in sequence Y_*.

The decoder first applies token embedding and positional encoding as

$$g^0_{u,1:k-1} = \mathrm{Embed}(y'_{u,1:k-1}) + \mathrm{PosEnc}(y'_{u,1:k-1}), \quad (6)$$

where $\mathrm{Embed}(\cdot)$ represents the token embedding. Next, the decoder computes hidden vector $g^n_{u,k-1}$ in each block n as

$$\bar{g}^{n-1}_{u,k-1} = \xi(g^{n-1}_{u,k-1}) \quad (7)$$

$$\tilde{g}^{n}_{u,k-1} = g^{n-1}_{u,k-1} + \mathrm{MHA}(\bar{g}^{n-1}_{u,k-1}, \bar{g}^{n-1}_{u,1:k-1}, \bar{g}^{n-1}_{u,1:k-1}) \quad (8)$$

$$\tilde{\tilde{g}}^{n}_{u,k-1} = \tilde{g}^{n}_{u,k-1} + \mathrm{MHA}(\xi(\tilde{g}^{n}_{u,k-1}), \bar{H}^{N_{\text{enc}}}_{v:u}, \bar{H}^{N_{\text{enc}}}_{v:u}) \quad (9)$$

$$g^{n}_{u,k-1} = \tilde{\tilde{g}}^{n}_{u,k-1} + \mathrm{FFN}(\xi(\tilde{\tilde{g}}^{n}_{u,k-1})), \quad (10)$$

and outputs the decoder states obtained as the normalized output of the last block, i.e., $\bar{g}^{N_{\text{dec}}}_{u,k-1} = \xi(g^{N_{\text{dec}}}_{u,k-1})$, where N_{dec} denotes the number of decoder blocks. Eq. (9) applies source attention over the encoder states, in which $\xi(\tilde{g}^{n}_{u,k-1})$ is used for the query vector. Finally, we obtain the Transformer token probability distribution by applying a linear transformation and a softmax function as

$$\begin{aligned}&p_{\text{trs}}(y_{u,i} | Y_{v:u-1}, y_{u,1:i-1}, X_{v:u}) \\ &\qquad = \mathrm{Softmax}(\mathrm{Linear}(\bar{g}^{N_{\text{dec}}}_{u,|Y_{v:u-1}|+i-1})). \quad (11)\end{aligned}$$

We can also utilize CTC in training and decoding similarly to the CTC-Attention approach in RNN-based architectures [6, 8, 28]. The CTC sequence probability can be computed as

$$p_{\text{ctc}}(Y_u | X_{v:u}) = \mathrm{CTC}(\mathrm{Softmax}(\mathrm{Linear}(\bar{H}^{N_{\text{enc}}}_{u})), Y_u), \quad (12)$$

where $\mathrm{CTC}(P, Y)$ is an operation that marginalizes the posterior probabilities over all possible alignments between P and Y using the forward-backward algorithm [1].

For training, we use the CTC-attention loss computed as

$$\begin{aligned}\mathcal{L}_u = &-\alpha \log p_{\text{trs}}(Y^*_u | Y^*_{v:u-1}, X_{v:u}) \\ &- (1-\alpha) \log p_{\text{ctc}}(Y^*_u | X_{v:u}), \quad (13)\end{aligned}$$

where Y^*_u and $Y^*_{v:u-1}$ are ground-truth transcripts, and α is a scaling factor to balance the Transformer and CTC losses.

For decoding, we combine Transformer, CTC, and optionally LM scores to find the best hypothesis as

$$\begin{aligned}\hat{Y}_u = &\underset{Y_u \in \mathcal{V}^*}{\operatorname{argmax}} \{\lambda \log p_{\text{trs}}(Y_u | Y_{v:u-1}, X_{v:u}) \\ &+ (1-\lambda) \log p_{\text{ctc}}(Y_u | X_{v:u}) + \gamma \log p_{\text{lm}}(Y_u)\}, \quad (14)\end{aligned}$$

where λ and γ are scaling factors to balance the model scores. Similarly to prior studies, we employ output-synchronous beam search to efficiently find the best hypothesis [7]. In Eq. (14), we can choose a context-dependent LM in a form of $p_{\text{lm}}(Y_u | Y_{1:u-1})$. With an RNN-LM, contextual information beyond one utterance can be used by passing the state information from the previous utterance. To recognize long audio recordings such as lecture and conversational speeches, we repeat the decoding process of Eq. (14) in a sliding-window fashion with one-utterance shifts [19].

3. Improvements to context-expanded Transformers

3.1. Context-expanded Conformer

Conformer is a variant of Transformers augmented with convolution [20], in which each encoder block has a convolution module right after the multi-head self-attention (MHSA) layer. The convolution module consists of layer normalization, point-wise convolution, gated linear unit (GLU) activation, 1D depth-wise convolution, batch normalization, Swish activation, point-wise convolution, and dropout. Besides, each MHSA layer in the encoder employs relative positional encoding [29], which allows the self-attention module to generalize better to different input length.The encoder block is also extended to have a sandwich structure, where the original feed-forward layer is replaced with two half-step feed-forward layers, one before MHSA and the other after the convolution module. It has been shown that the Conformer-based models provide substantial accuracy improvement over vanilla Transformers [20, 30].

The framework of context-expanded Transformer can be applied to the Conformer without changing its basic architecture. The encoder steps of Eqs. (4) and (5) are replaced with the sandwich structure including the convolution module as

$$\bar{\bar{H}}^{n}_{v:u} = H^{n-1}_{v:u} + \frac{1}{2}\mathrm{FFN}(\bar{H}^{n-1}_{v:u}) \tag{15}$$

$$\tilde{H}^{n}_{v:u} = \bar{\bar{H}}^{n}_{v:u} + \mathrm{MHA}(\xi(\bar{\bar{H}}^{n}_{v:u}), \xi(\bar{\bar{H}}^{n}_{v:u}), \xi(\bar{\bar{H}}^{n}_{v:u})) \tag{16}$$

$$\tilde{\tilde{H}}^{n}_{v:u} = \tilde{H}^{n}_{v:u} + \mathrm{Conv}(\xi(\tilde{H}^{n}_{v:u})) \tag{17}$$

$$H^{n}_{v:u} = \tilde{\tilde{H}}^{n}_{v:u} + \frac{1}{2}\mathrm{FFN}(\xi(\tilde{\tilde{H}}^{n}_{v:u})), \tag{18}$$

where $\mathrm{Conv}(\cdot)$ denotes the convolution module.

3.2. Efficient decoding with activation recycling

Computational complexity of multi-head attention increases quadratically with the sequence length. With context expansion, the input sequence is approximately 4 times longer than single utterances in a standard setting of context-expanded Transformer, obviously causing a major increase in decoding time.

Considering a sliding-window decoding with one-utterance shift moving focus to a new utterance X_u, instead of computing new hidden activation vectors for previous utterances $X_{v:u-1}$ taking into account the new current utterance X_u as in [19], we can reuse the hidden activation vectors computed for each previous utterance when it was decoded. Accordingly, the hidden activations in the encoder block can be computed as

$$\tilde{H}^{n}_{u} = H^{n-1}_{u} + \mathrm{MHA}(\bar{H}^{n-1}_{u}, \bar{H}^{n-1}_{v:u}, \bar{H}^{n-1}_{v:u}) \tag{19}$$

$$H^{n}_{u} = \tilde{H}^{n}_{u} + \mathrm{FFN}(\xi(\tilde{H}^{n}_{u})) \tag{20}$$

$$H^{n}_{v:u} = (\mathcal{H}^{n}_{v:u-1}, H^{n}_{u}), \tag{21}$$

where hidden activations $\mathcal{H}^{n}_{v:u-1}$ have already been computed and cached in the memory when the previous utterances were decoded. The activations thus need to be computed only for the current utterance X_u. Furthermore, in multi-head attention, the length of query is reduced from the segment length $|H^{n}_{v:u}|$ to the length of the current utterance $|H^{n}_{u}|$, i.e., the computational complexity is reduced from $O(|H^{n}_{v:u}|^2)$ to $O(|H^{n}_{u}| \times |H^{n}_{v:u}|)$.

The computation for other layers including feed-forward layers reduces to the same level as ordinary utterance-based ASR. For recycling the cached activations, it is important to employ relative positional encoding to make the activations independent of their positions. Moreover, as cached activations cannot have any information from future utterances, we need to omit backward self-attention across utterances by masking such connections during training.

This recycling mechanism can also be used for the decoder blocks, where relative positional encoding is also needed but the backward self-attention does not have to be considered since the decoder uses only forward self-attention. Furthermore, we can reduce the computation for source attention by limiting the attention span to only the current utterance, i.e., the computational complexity of source attention can be reduced from $O(|Y_{v:u}| \times |H^{N_{enc}}_{v:u}|)$ to $O(|Y_u| \times |H^{N_{enc}}_{u}|)$, which is the same as that of the utterance-based ASR. The same recycling mechanism can be applied for context-expanded Conformers as well.

3.3. Streaming decoding with triggered attention

Streaming decoding is regarded as an essential function in ASR systems. However, most end-to-end models are designed to access the full input sequence even for predicting the first token. This means that ASR decoding cannot start until the utterance end point is detected, leading to a substantial delay in the ASR output especially for long utterances. Recent studies have proposed efficient streaming algorithms for end-to-end ASR [21–23, 31–33], which enable processing of the input speech in a streaming fashion and reduce the latency. For joint CTC-attention-based models, we have proposed the triggered attention technique [21–23], which utilizes CTC during training and inference to estimate a token emission timing and activate the attention decoder accordingly. Triggered attention provides time-synchronous decoding using CTC prefix beam search extended with on-the-fly attention decoder rescoring.

In this paper, we introduce triggered attention into context-expanded Transformers to realize high-accuracy streaming ASR. The training is performed on concatenated utterances and enforces time restriction on the self- and source-attention layers by masking attention weights to simulate a situation where future context is not available while still considering several look-ahead frames. Activation recycling is also employed in the decoding process.

3.4. Related work

Apart from our own prior work [19], prior studies on end-to-end ASR for long-form speeches [16–18, 32] rely on attention-based encoder-decoders or RNN-T models, while we here explore the use of Transformer-based encoder-decoders and their extensions.

Regarding the activation recycling technique, a similar method has already been introduced in the Transformer-XL language model (LM) [29]. However, our Transformer has an encoder-decoder architecture and reuses activations on an utterance basis rather than fixed-sized text blocks as in Transformer-XL, since utterances are suitable as a basic processing unit for ASR. Moreover, we forbid backward self-attention across utterances in the encoder and apply within-utterance source attention in the decoder during training, both of which have not been considered in Transformer-XL.

Streaming decoding is an active area of research in end-to-end ASR [31–33]. One prior work did explore a streaming technique for long-form ASR with RNN-T [32], in which the decoding method depends on the RNN architecture, where the encoder and decoder receive state information from the previous utterance, but that technique is not applicable to Transformers.

4. Experiments

4.1. Experimental setup

We conducted several experiments using conversational ASR benchmarks on the HKUST [24] and Switchboard [25] corpora, which consist of 200 hours and 300 hours of 8 kHz telephone conversations in Mandarin Chinese and English, respectively.

The Kaldi toolkit [34] was used to extract 80-dimensional log mel-filter bank acoustic features plus three-dimensional pitch features. We trained Transformers with the architecture

Table 1: *Recognition error rate vs. decoding time in HKUST and Switchboard benchmarks. The decoding time is represented in real-time factor (RTF), when decoded with a single-thread process on an Intel Core i7-3970X CPU @ 3.50GHz.*

	HKUST		Switchboard	
	dev		CH / SWB	
	CER [%]	RTF	WER [%]	RTF
Baseline Transformer	21.2	0.44	15.5 / 7.8	0.66
Context-Ex. Transformer	18.9	1.26	14.0 / 7.1	1.89
+ Activation recycling	**18.8**	**0.61**	**14.0** / 7.2	**0.93**
Baseline Conformer	20.0	0.46	13.9 / 6.8	0.68
Context-Ex. Conformer	17.3	1.31	12.1 / 6.3	2.05
+ Activation recycling	**17.3**	**0.62**	**12.0 / 6.3**	**1.02**
ESPnet Conformer [30]	22.2	-	15.0 / 7.1	-
Att. enc-dec. [36]	-	-	14.0 / 6.8	-
Att. enc-dec. [37]	-	-	12.5 / 6.4	-

in ESPnet [11, 35]. The encoder had one Conv2D module followed by 12 encoder blocks ($N_{\rm enc} = 12$). The Conv2D included a 2-layer 2D-CNN with 256 channels, a kernel size of 3×3, a stride of size 2, and ReLU activation, which outputs a 256-dimensional vector sequence with the utterance length reduced by a factor of 4. We employed multi-head attention with 4 heads of 256 dimensions. The feed-forward network had one hidden layer with 2,048 units and ReLU non-linearity. The decoder had a token embedding layer followed by 6 decoder blocks ($N_{\rm dec} = 6$). The self-attention, source attention, and feed-forward layers in the decoder had the same dimensions as those in the encoder. The output dimension was dependent on the number of unique tokens in the task, with 3,653 characters in HKUST and 1,996 word pieces in Switchboard.

We basically followed the default configuration of ESPnet recipes [35], where speed perturbation and SpecAugment [36] were applied for both data sets. Baseline Transformers were trained with independent utterances without context. To train Transformers with the proposed method, we expanded each utterance to a 20-second segment by concatenating it with previous utterances. Unlike our prior work [19], we trained the context-expanded models with CTC-attention loss in Eq. (13) by fine-tuning a corresponding utterance-based model using Adam optimizer, since our preliminary experiments indicated that fine-tuning is more stable and faster than training from scratch as we have done before. Finally, we averaged the top 10 models based on validation accuracy for recognition.

We also trained RNN-LMs using transcripts for HKUST and Switchboard, further adding transcripts from the Fisher corpus for Switchboard. The LMs had 2 LSTM layers with 650 cells for HKUST and 1,024 cells for Switchboard. The transcripts were concatenated in the same manner as in context-expanded Transformer training. ASR performance was measured by character error rate (CER) or word error rate (WER).

4.2. Results

Table 1 shows recognition error rate and decoding speed for utterance-based (Baseline) and context-expanded (Context-Ex.) models. For both datasets, context expansion provides substantial relative error reduction ranging from 5% to 13.5% for both Transformers and Conformers. The context-expanded Conformer achieved 17.3% CER on the HKUST dev set and 12.0% / 6.3% WER on the CallHome (CH) and Switchboard (SWB) subsets of the Switchboard Eval2000 test set. These numbers are better than those reported in other papers [30, 36, 37]. We also evaluated the effect of activation recycling in beam search decoding [38] with beam size 10. In decoding, we increased the segment size to 25 seconds for context-expanded models on Switchboard, since this provides slightly better WERs for the validation and test sets. Without recycling, the RTF increases roughly by a factor of 3 from the baseline. With recycling, however, the decoding time can be reduced to half of the original time with almost no increase of errors. The recycling mechanism thus effectively works for context-expanded Transformers. Consequently, we can perform context-expanded ASR with a 35-50% increase of computation time from the baseline on a single CPU core.

Table 2: *Streaming ASR results. Numbers indicate CERs [%] for HKUST and WERs [%] for Switchboard. "%inc" indicates the error increase ratio from the full-sequence model.*

	HKUST	Switchboard	
	dev (%inc)	CH (%inc)	SWB (%inc)
Baseline Transformer	21.2	15.5	7.8
+ Triggered attention	23.1 (9.0)	17.9 (15.5)	9.0 (15.4)
Context-Ex. Transformer	18.8	14.0	7.2
+ Triggered attention	**20.2 (7.4)**	**15.5 (10.7)**	**8.4** (16.7)
Baseline Conformer	20.0	13.9	6.8
+ Triggered attention	22.6 (13.0)	16.8 (20.0)	8.0 (17.6)
Context-Ex. Conformer	17.3	12.0	6.3
+ Triggered attention	**19.3 (11.6)**	**14.2 (18.3)**	**7.2 (14.3)**

Next, we investigate streaming ASR with triggered attention. In triggered attention decoding, we applied a 1-frame self-attention look-ahead at each encoder layer, which results in a 12-frame delay for the 12-layer encoder. We also used a 12-frame look-ahead on the encoder frames in source attention by the decoder. Accordingly, this configuration requires a 480 ms + 480 ms theoretical delay, since one encoder frame corresponds to 40 ms. For Conformers, we restricted the depth-wise convolution to use only past frames. The error rates are summarized in Table 2. The results show that the context-expanded models can reduce the errors in streaming ASR as well. We also observe a slight deterioration of error rate from full-sequence (i.e., not streaming) ASR, as is usually observed due to a lack of future information in streaming ASR. But, according to the error increase ratio (%inc), the context-expanded models tend to mitigate the increase in errors compared to the baseline models. This fact suggests that the context-expanded Transformers are more effective in streaming ASR, since the utterance-based models can use only little information at an early stage of decoding while the context-expanded models can utilize more contextual information from previous utterances.

5. Conclusions

In this paper, we have extended our prior work on context-expanded Transformers, which exploit contextual information of previous utterances to improve ASR accuracy for the current utterance. We have investigated three extensions: (1) Conformer architecture for further accuracy improvement, (2) accelerated decoding by activation recycling, and (3) streaming decoding with triggered attention. We have demonstrated that the extended Transformers provide state-of-the-art end-to-end ASR performance, achieving 17.3% CER on the HKUST dev set and 12.0% / 6.3% WER on the Switchboard-300 Eval2000 CallHome/Switchboard test sets. The new decoding method reduced decoding time to under 50% of that of the original method and further enabled streaming ASR without considerable accuracy degradation.

6. References

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

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[3] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE ICASSP*, Apr. 2015.

[4] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE ICASSP*, May 2013, pp. 6645–6649.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, Dec. 2017, pp. 5998–6008.

[6] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. IEEE ICASSP*, Mar. 2017, pp. 4835–4839.

[7] 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 *Proc. ISCA Interspeech*, Aug. 2017.

[8] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," in *Proc. ISCA Interspeech*, Sep. 2019, pp. 1408–1412.

[9] 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," in *Proc. IEEE ICASSP*, May 2020, pp. 7829–7833.

[10] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE ICASSP*, Apr. 2018, pp. 5884–5888.

[11] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs RNN in speech applications," in *Proc. IEEE ASRU*, Dec. 2019.

[12] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, 2010.

[13] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for English conversational speech recognition," *arXiv preprint arXiv:1703.07754*, 2017.

[14] Z. Fan, J. Li, S. Zhou, and B. Xu, "Speaker-aware speech-transformer," in *Proc. IEEE ASRU*, Dec. 2019, pp. 222–229.

[15] L. Sarı, N. Moritz, T. Hori, and J. Le Roux, "Unsupervised speaker adaptation using attention-based speaker memory for end-to-end ASR," in *Proc. IEEE ICASSP*, May 2020, pp. 7384–7388.

[16] S. Kim and F. Metze, "Dialog-context aware end-to-end speech recognition," in *Proc. IEEE SLT*, Dec. 2018, pp. 434–440.

[17] R. Masumura, T. Tanaka, T. Moriya, Y. Shinohara, T. Oba, and Y. Aono, "Large context end-to-end automatic speech recognition via extension of hierarchical recurrent encoder-decoder models," in *Proc. IEEE ICASSP*, May 2019, pp. 5661–5665.

[18] C.-C. Chiu, W. Han, Y. Zhang, R. Pang, S. Kishchenko, P. Nguyen, A. Narayanan, H. Liao, S. Zhang, A. Kannan *et al.*, "A comparison of end-to-end models for long-form speech recognition," *arXiv preprint arXiv:1911.02242*, 2019.

[19] T. Hori, N. Moritz, C. Hori, and J. Le Roux, "Transformer-based long-context end-to-end speech recognition," in *Proc. ISCA Interspeech*, Oct. 2020, pp. 5011–5015.

[20] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. ISCA Interspeech*, Oct. 2020, pp. 5036–5040.

[21] N. Moritz, T. Hori, and J. Le Roux, "Triggered attention for end-to-end speech recognition," in *Proc. IEEE ICASSP*, May 2019, pp. 5666–5670.

[22] N. Moritz, T. Hori, and J. Le Roux, "Streaming end-to-end speech recognition with joint CTC-attention based models," in *Proc. IEEE ASRU*, Dec. 2019, pp. 936–943.

[23] N. Moritz, T. Hori, and J. Le Roux, "Streaming automatic speech recognition with the transformer model," in *Proc. IEEE ICASSP*, May 2020.

[24] Y. Liu, P. Fung, Y. Yang, C. Cieri, S. Huang, and D. Graff, "HKUST/MTS: A very large scale mandarin telephone speech corpus," in *Chinese Spoken Language Processing*. Springer, 2006, pp. 724–735.

[25] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: telephone speech corpus for research and development," in *Proc. IEEE ICASSP*, vol. 1, 1992, pp. 517–520.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Jun. 2016, pp. 770–778.

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

[28] T. Hori, S. Watanabe, and J. R. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proc. ACL*, Jul. 2017.

[29] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. ACL*, Jul. 2019, pp. 2978–2988.

[30] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi *et al.*, "Recent developments on ESPnet toolkit boosted by Conformer," *arXiv preprint arXiv:2010.13956*, 2020.

[31] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *Proc. IEEE ICASSP*, May 2019, pp. 6381–6385.

[32] A. Narayanan, R. Prabhavalkar, C.-C. Chiu, D. Rybach, T. N. Sainath, and T. Strohman, "Recognizing long-form speech using streaming end-to-end models," *arXiv preprint arXiv:1910.11455*, 2019.

[33] E. Tsunoo, Y. Kashiwagi, and S. Watanabe, "Streaming Transformer ASR with blockwise synchronous beam search," in *Proc. IEEE SLT*, Jan. 2021, pp. 22–29.

[34] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. IEEE ASRU*, Dec. 2011.

[35] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N.-E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. ISCA Interspeech*, Sep. 2018.

[36] 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. ISCA Interspeech*, Sep. 2019, pp. 2613–2617.

[37] Z. Tüske, G. Saon, K. Audhkhasi, and B. Kingsbury, "Single headed attention based sequence-to-sequence model for state-of-the-art results on Switchboard-300," in *Proc. ISCA Interspeech*, Oct. 2020.

[38] H. Seki, T. Hori, S. Watanabe, N. Moritz, and J. Le Roux, "Vectorized beam search for CTC-attention-based speech recognition," in *Proc. ISCA Interspeech*, Sep. 2019, pp. 3825–3829.