INTERSPEECH 2021: Show & Tell Contribution
30 August – 3 September, 2021, Brno, Czechia

On-device Streaming Transformer-based End-to-End Speech Recognition

Yoo Rhee Oh, Kiyoung Park *

Artificial Intelligence Research Laboratory, ETRI, Daejeon, Republic of Korea
yroh@etri.re.kr, pkyoung@etri.re.kr

Abstract

This work is the first attempt to run streaming Transformer-based end-to-end speech recognition on embedded scale IoT systems. Recently there are many researches on online Transformer-based speech recognition such as a contextual block encoder [1] and a block-wise synchronous beam search [2]. Based on them we designed a novel fully-streaming end-to-end speech recognition method using Transformer. By efficiently utilizing a connectionist temporal classification network to detect symbol and sentence boundaries, we make decoder in streaming manner. Moreover, by using the optimized model structure, the proposed method could be deployed on a low-power edge device such as Raspberry Pi 4B with the high accuracy and the small latency. With the experiments with Librispeech corpus, the methods achieved word error rates of 3.76% and 9.25% respectively. Also the recognition speed is measured in two aspects; the real-time factor and the user perceived latency. The system is evaluated to have 0.84 xRT and the average latency of 0.75±0.62 seconds on Raspberry Pi 4B.

Index Terms: speech recognition, Transformer, streaming, end-to-end

1. Introduction

Not long after automatic speech recognition(ASR) using end-to-end mechanism is firstly introduced, the technique is developed so fast that it is recording the state-of-the-art performance for many benchmark tests. Among them the transformer and the conformer are leading the scoreboard. One of the merits of those models is that the sequence to decode are fed to the model at once and processed in parallel using self-attention, hence the relation in the input sequence is well represented by the encoder, and utilized by the decoder in the model. However this imposes the limitation that decoding can only be started after the entire sequence is fed to the model. For the applications where the instant ASR results are required such as real-time subtitle generation, ASR-aided stenography, etc, this limitation is the key problem to be overcome.

Recently there are many researches in streaming end-to-end ASR based on recurrent neural network transducer(RNN-T) model [3], listen-attend-spell(LAS) model [4] and Transformer [2]. In this paper we present fully streaming Transformer which can run on low profile embeded board. We trained the model using ESPNet [5] with the model configuration which is suitable for low profile streaming ASR. The efficient streaming decoding algorithm is proposed by incorporating end-of-block and end-of-speech detection with connectionist temporal classfication(CTC) algorithm. The resultant system can decode speech in real-time on the Raspberry Pi 4 Model B which has Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz using about 350MB of memory.

* Corresponding author

2. System description

Fig. 1 illustrates the overview of the on-device streaming CTC/Transformer-based end-to-end ASR system. The system consists of a block-wise feature extraction, a contextual Transformer encoder [1], and a block-wise beam search.

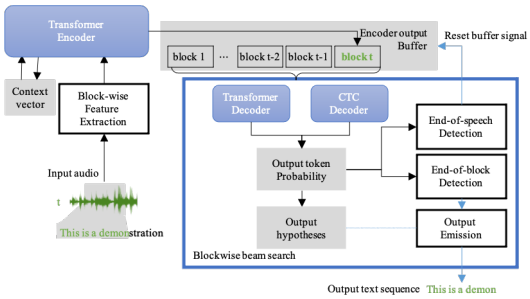

Figure 1: *Overview of the on-device streaming CTC/Transformer-based end-to-end ASR system. Blue- and gray-colored blocks: the models and the variable data. Blue line: an optional path according to a decision.*

2.1. Feature extraction

For each incoming audio block, the 80-dimensional log-Mel filterbank coefficients are extracted for every 10–ms analysis frame and the features are normalized using a global mean and variance. The block size is the hop size of the contextual Transformer encoder.

2.2. A contextual Transformer encoder

For the encoder to work in streaming way, the input features are processed in block-wise [1]. Instead of utilizing all previous sequence of blocks, a context embedding vector is computed over a block, and then passed to the next block. The context embedding vectors for all the encoder layers are buffered and passed to the next block. In this work, deeper encoder(24 layers) is used to compensate the reduced number of layers in decoder(2 layers).

2.3. Block-wise beam search

A block of encoder output is decoded into the text sequence up to the time via a block-wise beam search. The block-wise beam search is extended from the method of [2]. It consists of Transformer and CTC decoders and end-of-speech and end-of-block detectors. In brief, for the given encoded blocks, a new output token is estimated using the decoders so as to sequentially generate the output text. The text generation is repeated until end-of-block or end-of-speech is detected. The brief explanations for 4 components in this module are as followings:

Table 1: *Word error rates (%), real-time factor and user perceived latency of streaming ASR systems employing the sequence-to-sequence methods on Librispeech with no language model.*

Method	clean	other	Avg.	RTF	Latency
SAGMM-tr [10]	3.67	9.76	6.72	-	-
CBP-ENC+BBD [2]	4.27	9.66	6.97	-	-
Proposed system	3.76	9.25	6.51	0.84	0.75±0.62

Transformer decoder: The autoregressive behavior of the decoder leads to high computations. Therefore, this work uses swallower decoder of 2 layers for an on-device deployment.

CTC decoder: The CTC decoder improves the performance of a Transformer-based ASR [6]. And it properly detects end-of-block for low-latency streaming ASR.

End-of-speech detector: It decides when to free the resources. We utilize two end-of-speech detections methods [7] [8]. For long speech, duration- and token-length-based triggering are used.

End-of-boundary detector: It is used to decide when to emit the text sequence for the encoded blocks. The block boundary is detected based on the probabilities of hypotheses.

3. Experiments

The streaming ASR system was evaluated using Librispeech corpus [9].

3.1. Accuracy

SAGMM-tr [10] is the state-of-the-art streaming sequence-to-sequence method at the time of the submission. CBP-ENC+BBD is the streaming Transformer-based ASR which motivated this work. As shown in Table 1, the proposed streaming ASR system achieved better accuracy than SAGMM-tr. Moreover, the performance of the proposed ASR system is comparable to the state-of-the-art Transducer-based streaming method [11].

3.2. Recognition speed

To measure the recognition speed simulating real-world applications, we prepared longer test data by concatenating utterances from the same speaker in libriclean dataset not to exceed 1 minute in length. 10 utterances are randomly selected from concatenated dataset and fed into ASR system running on the Raspberry Pi 4B. Two aspects of speed are measured; the real-time factor(RTF) and the word-wise user perceived latency.

RTF is measured as the ratio of the elapsed time to decode an utterance to the length of it. The lower is the faster and the value should be less than 1 for the system to operate in real-time. To measure the elapsed time, the ASR fetches data from file system and decode it as fast as it can.

Apart from the RTF, the latency is another important measure to evaluate the speed of ASR system. We define word-wise user perceived latency as the time elapsed to get the result of ASR for each word after the word is spoken. To measure it, we fed the test speech to the system via line-in as if it is being uttered by human, and check the time at which each of the output words is emitted. The time at which those words are spoken is estimated by time-aligning the resulting recognition result to the recorded speech signal. The same data which is used to measure RTF is used to measure the latency. The average values of

Figure 2: *Snapshot of the streaming CTC/Transformer-based ASR system on Raspberry Pi 4B.*

RTF and user perceived latency for 10 utterances are shown in Table. 1

4. Demonstration and conclusions

We will demonstrate three streaming CTC/Transformer-based ASR systems deployed on Raspberry Pi 4B, as shown in Fig. 2: (a) English ASR system trained with Librispeech and (b) two ASR systems trained with large-corpus English and Korean. We believe that this work will contribute to encourage or advance the researches on a streaming Transformer-based ASR.

5. Acknowledgements

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2019-0-01376, Development of the multi-speaker conversational speech recognition technology)

6. References

[1] E. Tsunoo, Y. Kashiwagi, T. Kumakura, and S. Watanabe, "Transformer ASR with Contextual Block Processing," in *Proc. of ASRU*, 2019, pp. 427–433.

[2] E. Tsunoo, Y. Kashiwagi, and S. Watanabe, "Streaming Transformer ASR With Blockwise Synchronous Beam Search," in *Proc. of SLT*, 2021, pp. 22–29.

[3] Y. H. et al., "Streaming End-to-end Speech Recognition for Mobile Devices," in *Proc. of ICASSP*, 2019, pp. 6381–6385.

[4] R. Hsiao, D. Can, T. Ng, R. Travadi, and A. Ghoshal, "Online Automatic Speech Recognition With Listen, Attend and Spell Model," *IEEE Signal Processing Letters*, vol. 27, pp. 1889–1893, 2020.

[5] S. W. *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. of INTERSPEECH*, 2018, pp. 2207–2211.

[6] S. K. *et al.*, "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration," in *Proc. of INTERSPEECH*, 2019, pp. 1408–1412.

[7] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[8] Y. R. Oh, K. Park, and J. Park, "Transformer-based ASR using multiple-utterance beam-search," 2021.

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

[10] T. G. K. *et al.*, "Learning Monotonic Alignments with Source-Aware GMM Attention," 2021.

[11] J. Y. *et al.*, "Dual-mode ASR: Unify and Improve Streaming ASR with Full-context Modeling," in *Proc. of ICLR*, 2021.