Blockwise Streaming Transformer for Spoken Language Understanding and Simultaneous Speech Translation

Keqi Deng¹, Shinji Watanabe², Jiatong Shi², Siddhant Arora²

¹University of Chinese Academy of Sciences, China
²Carnegie Mellon University, USA

dengkeqi20@mails.ucas.ac.cn, shinjiw@cmu.edu, {jiatongs, siddhana}@cs.cmu.edu

Abstract

Although Transformers have gained success in several speech processing tasks like spoken language understanding (SLU) and speech translation (ST), achieving online processing while keeping competitive performance is still essential for real-world interaction. In this paper, we take the first step on streaming SLU and simultaneous ST using a blockwise streaming Transformer, which is based on contextual block processing and blockwise synchronous beam search. Furthermore, we design an automatic speech recognition (ASR)-based intermediate loss regularization for the streaming SLU task to improve the classification performance further. As for the simultaneous ST task, we propose a cross-lingual encoding method, which employs a CTC branch optimized with target language translations. In addition, the CTC translation output is also used to refine the search space with CTC prefix score, achieving joint CTC/attention simultaneous translation for the first time. Experiments for SLU are conducted on FSC and SLURP corpora, while the ST task is evaluated on Fisher-CallHome Spanish and MuST-C En-De corpora. Experimental results show that the blockwise streaming Transformer achieves competitive results compared to offline models, especially with our proposed methods that further yield a 2.4% accuracy gain on the SLU task and a 4.3 BLEU gain on the ST task over streaming baselines.

Index Terms: streaming Transformer, spoken language understanding, speech translation

1. Introduction

In the last decade, deep learning has greatly promoted the development of several speech processing tasks like spoken language understanding (SLU) [1] and speech translation (ST) [2]. SLU task aims to extract structured semantic representations from speech signals [3, 4]. Conventional cascaded SLU systems consist of an automatic speech recognition (ASR) module as well as a downstream natural language understanding (NLU) module [5]. On the other hand, end-to-end (E2E) SLU systems directly extract users’ intentions from input speech to avoid error propagation seen in the above cascaded method [1, 6]. Similarly, E2E ST systems directly translate source language speech into target language text and have advantages such as lower latency, smaller model size, and less error compounding over cascaded ST [7, 8]. However, for real-world human-computer interactions, it is still essential to make the systems online while keeping competitive performance.

Online systems effectively reduce the processing latency by in-time responses before consuming the full input speech [4]. For streaming SLU task, [4] takes the first step to achieve an E2E streaming SLU model based on connectionist temporal classification (CTC) objective [9]. [10] further employs an adapted version, named connectionist temporal localization, for online SLU. These approaches identify intent when sufficient evidence has been accumulated, without waiting until the end of the utterance. As for the simultaneous ST (SST) task, recent works can be divided into two categories: fixed policy and flexible policy [11]. Several works employ fixed policy to SST [12] by adapting the text-based wait-k strategy [13], while we also observe works with flexible policies through monotonic attention [14] and other variants [15, 16].

However, unlike simultaneous text translation, whose input is already segmented into words, the wait-k strategy faces a challenge of computing the number of valid tokens for a source speech when applied to the SST task [17]. And flexible policy methods like MoChA [16] significantly degrade translation quality and make it difficult to keep an acceptable latency [18]. In addition, the current decoding process of ST is mostly based on attention-based inference, which suffers from poor text length prediction [19]. Meanwhile, on streaming SLU tasks, although the CTC model has been proved effective, online systems based on popular encoder-decoder structures are still worth exploring.

In this paper, we propose to use blockwise streaming Transformer [18], which has been proved to outperform prior methods like MoChA in the ASR task, for streaming SLU and simultaneous ST systems. Inspired by [20], we further design an ASR-based intermediate loss to help the model better converge for streaming SLU. As for the simultaneous ST, we propose a cross-lingual encoding (CLE) method by injecting a CTC objective between encoder outputs and target translations. In addition, the CTC-based alignment is also used to refine the search space via considering the CTC prefix scores, by which we achieve a joint CTC/attention simultaneous translation for the first time. Experimental results show that the blockwise streaming Transformer originally developed for ASR can achieve competitive results on SLU and ST tasks. And our proposed methods can further yield a 2.4% accuracy gain on the streaming SLU task and a 4.3 BLEU gain on the simultaneous ST task.

2. Blockwise Streaming Transformer

Block processing is an effective way to make the Transformer encoder online [21, 22, 23]. As shown in Fig. 1, [22] introduces a context inheritance mechanism to utilize richer contextual information. Previous context is encoded into context embedding, which is calculated for each block at each sublayer and then handed over to the next sublayer. We denote the i-th block of encoded feature as $B^i = (B_{i1}, \ldots, B_{iL})$.

To achieve streaming decoding, [18] further proposes a blockwise synchronous beam search based on the contextual

* We release this streaming ST/SLU publicly available through the ESPnet open source toolkit.
block processing [22]. The decoder predicts next word $y_t$ based on previous output $y_{t-1}$ and block feature $B$.
In addition, a CTC prefix score is also computed based on $B$ to achieve joint CTC/attention decoding [18, 24]. When a hypothesis contains an end token or a repetition, this prediction is regarded as unreliable with insufficient $b$ blocks, and then the decoder waits for the next block to be encoded [18].

3. Proposed Method

In this paper, we propose to use a blockwise streaming Transformer for the streaming SLU and simultaneous ST tasks.

3.1. Streaming spoken language understanding

E2E SLU combines ASR and NLU into one task [4], thus requiring both acoustic and semantic understanding [6]. Prior work observes that adding auxiliary ASR objectives by training models to predict both intent and transcript can improve the performance of SLU systems [6].

However, unlike the ASR token, the intent has no corresponding speech frames, which makes it hard for the mono- tone alignment model like CTC to learn the target that contains both intent and ASR transcripts. Therefore, we argue that a model should first distinguish words before learning to understand intents. Under the framework of blockwise streaming Transformer, we use an ASR-based intermediate loss regularization method to promote the learning process, which is shown in Fig. 2. We apply an extra CTC branch to the $M$-th encoder layer and an auxiliary CTC loss is computed with ASR transcripts as the target. In this way, lower layers of the encoder are encouraged to distinguish different tokens, while the higher encoder layers try to understand semantics for intent classification. The final training objective $L_{	ext{fin}}$ is calculated as follows:

$$L_{	ext{fin}} = \lambda L_{\text{ctc}} + L_{\text{aux}} + (1 - \lambda) L_{\text{ce}},$$

where $\lambda \in [0, 1]$, $L_{\text{ctc}}$ is the main CTC loss, $L_{\text{aux}}$ denotes the auxiliary loss, and $L_{\text{ce}}$ represents the cross-entropy (CE) loss. This extra CTC branch is discarded during inference thus does not break the online algorithms of streaming Transformer.

3.2. Simultaneous speech translation

E2E ST combines ASR and machine translation (MT) into a single task [25]. The optimization of E2E ST models can be more difficult than individually training ASR and MT models [26]. Multi-task learning and pre-training methods from ASR tasks are always used to alleviate the problem [26, 27].

Current works on ST mostly rely on attention-based decoding, which results in poor generation due to wrong text length [19]. In this paper, we propose a cross-lingual encoding method that Transformer with CTC as the objective has reordering capability [28]. As for simultaneous ST task, the reordering capability of the Transformer is somewhat limited, but the context inheritance mechanism of blockwise streaming Transformer is capable of encoding the context of previously processed blocks [22]. Therefore, we believe that CTC is still feasible and helps to refine the search space during decoding.

During training, with the target language translations as the learning target, we calculate a CTC loss from the CTC branch applied after the encoder and a CE loss from the decoder. In addition, to help the encoder converge better, we also apply an ASR-based intermediate CTC loss for ASR multi-task learning. The final training objective $L_{\text{fin}}$ is computed as follows, where $L_{\text{ctc}}$ and $L_{\text{ce}}$ are for ST task while $L_{\text{aux}}$ is for the auxiliary ASR task. $\gamma$ and $\beta$ are weights of CTC loss for ST and ASR, respectively.

$$L_{\text{fin}} = (1 - \gamma)(1 - \beta)L_{\text{cte}} + \beta L_{\text{cte}} + \gamma L_{\text{aux}}$$

3.2.1. Cross-lingual encoding (CLE)

Our proposed cross-lingual encoding (CLE) method employs a CTC branch optimized with target language translations. Although the CTC’s success on the ST task is counter-intuitive due to its monotonic property, previous works [28, 29] have proved that Transformer with CTC as the objective has reordering capability [28]. As for simultaneous ST task, the reordering capability of the Transformer is somewhat limited, but the context inheritance mechanism of blockwise streaming Transformer is capable of encoding the context of previously processed blocks [22]. Therefore, we believe that CTC is still feasible and helps to refine the search space during decoding.

3.2.2. Joint CTC/attention simultaneous translation

Attention-based encoder-decoder (AED) has become the most popular structure for the E2E ST task [30, 31]. The AED system solves the ST task as a sequence mapping and utilizes an attention mechanism [32] to achieve alignments between acoustic inputs and translated tokens. However, the attention mechanism mainly relies on the dependency between decoder states to decide whether to stop, thus having a flaw in poor generation performance due to wrong text length [19]. Furthermore, in the E2E ST task, the length mismatch between acoustic input and translated tokens is distinct and varies greatly from case to case, making it harder to track the attention-based alignments [33].

Therefore, we aim to utilize the CTC output to refine the search space and eliminate irregular alignments during decoding.

2Strictly speaking, CLE may have limited effect for language pairs with very serious reordering issue, but in this case, achieving simultaneous ST itself is a very hard topic. Therefore, we believe that our proposed CLE is feasible for selected simultaneous ST scenarios like English to German.

Figure 1: Illustration of the contextual block processing method.

Figure 2: Illustration of the ASR-based intermediate loss regularization, which is denoted as auxiliary CTC loss.
Table 1: BLEU and average lagging (AL) (ms) of different ST systems on Fisher-CallHome Spanish corpus. Details about the block size and cross-lingual encoding (CLE) are respectively introduced in Section 2 and Section 3.2.1.

| Model                  | Cross-lingual Encoding | Block Size | Decoding Style                  | Fisher | CallHome |
|------------------------|------------------------|------------|---------------------------------|--------|----------|
|                        |                        |            |                                 | dev    | dev2     |
| Offline Transformer    | √                      | √          | Attention-based beam search     | 44.5   | 45.3     |
| + ASR encoder init.    |                        |            |                                 | 48.2   | 48.1     |
| Streaming Transformer  | √                      | 20         | Attention-based beam search     | 40.6   | 41.5     |
| + ASR encoder init.    | √                      | 20         |                                 | 44.4   | 45.4     |
| Streaming Transformer  | √                      | 20         | Attention-based beam search     | 40.9   | 40.9     |
| + ASR encoder init.    | √                      | 20         |                                 | 45.2   | 45.4     |
| Streaming Transformer  | √                      | 20         | CTC/attention joint translation | 43.6   | 44.1     |
| + ASR encoder init.    | √                      | 20         |                                 | 47.4   | 47.9     |
| Streaming Transformer  | ×                      | 40         | Attention-based beam search     | 41.0   | 42.0     |
| + ASR encoder init.    | ×                      | 40         |                                 | 44.6   | 45.7     |
| Streaming Transformer  | √                      | 40         | Attention-based beam search     | 41.1   | 41.2     |
| + ASR encoder init.    | √                      | 40         |                                 | 45.4   | 46.5     |
| Streaming Transformer  | √                      | 40         | CTC/attention joint translation | 43.6   | 43.8     |
| + ASR encoder init.    | √                      | 40         |                                 | 47.9   | 48.2     |

Table 1: Illustration of the proposed joint CTC/attention simultaneous translation.

4. Experiments

4.1. Corpus

We evaluate our SLU systems on Fluent Speech Commands (FSC) [34] and Spoken Language Understanding Resource Package (SLURP) [35] corpora. We use the official train, dev, and test sets for all our experiments.

As for the ST task, the experiments are conducted on Fisher-CallHome Spanish [36] and Must-C En-De [37] corpora. We report case-insensitive BLEU. Fisher-CallHome Spanish corpus consists of 170 hours Spanish speech, the Spanish transcripts, and the English translations. Must-C En-De contains English speech collected from TED talks, the English transcripts, and the German translations.

4.2. Model descriptions

We used ESPnet2 toolkit [38] to build the models. For acoust-ic input, we employ 80-dimensional filter banks. For text output of the SLU task, we use 127 word units with 31 intents for FSC and 498 word units with 69 intents for SLURP. As for the ST task, for both source and target language, we employ sub-words based on BPE [39] with 500 and 4000 sizes for Fisher-CallHome Spanish and Must-C En-De corpora, respectively.

We develop offline Transformer baseline models following the recipe of ESPnet2 [38], which employs multi-task learning from ASR task (i.e., an ASR CTC branch applied after the encoder). Our streaming Transformer models also have a 12-layer encode and a 6-layer decoder, in which the attention dimension, feed-forward dimension, and attention heads are kept the same as the offline models. And the ASR-based intermediate loss is computed from the 8-th encoder layer. For block processing [18], we keep both hop size and look-ahead size accounting for 20% of the block size (after down-sampling). As for Must-C En-De, both the streaming and offline Transformer models have 256 attention dimensions, 2048 feed-forward dimensions, and 4 heads. During streaming decoding, our implementation is based on 640 ms simulated chunk 4. For simultaneous ST task, we evaluate the latency with average lagging (AL) defined in [11] on the evltest of CallHome set or the tst-COMMON of Must-C. As for streaming SLU task, we report endpoint latency (EP) [40] using a TITAN RTX GPU.

To prevent overfitting, we adopt a model averaging method. SpecAugment [41] is also used. A in Eq. 1 is set to 0.3, while \( \beta \) and \( \gamma \) in Eq. 2 are also set to 0.3. During decoding, for the SLU task, the CTC weight is set to 0.5; as for the ST task, the CTC weight \( \mu \) in Eq. 4 is set to 0.3 if the joint CTC/attention simultaneous translation is used. The beam size is 10.

\[ \text{CTC Branch} \]

\[ \text{Decoder} \]

\[ \text{Previous tokens} \]

\[ \text{Source Input} \]

\[ \text{Joint CTC/attention Translation} \]

\[ \text{Encoder} \]

Figure 3: Illustration of the proposed joint CTC/attention simultaneous translation.

---

4 The ESPnet2 example on this corpus is not yet open source in the current version of ESPnet.

4 https://github.com/espnet/espnet/blob/master/espnet2/bin/asr_inference_streaming.py
4.3. Simultaneous ST results

We compare our proposed simultaneous ST systems with the offline ST systems and conduct ablation studies to verify the effectiveness of our proposed CLE and CTC/attention joint translation methods. The experimental results on Fisher-CallHome Spanish are shown in Table 1, where ASR encoder init. means to pre-train the encoder with ASR task [26]. The results show that the ASR pre-training works for both the offline and streaming models. In addition, the BLEU of our simultaneous ST system increases slightly with larger block size, while the latency (e.g. AL) also increases. Furthermore, with our proposed Cross-lingual encoding method, the streaming Transformer ST model can choose joint CTC/attention simultaneous translation thus greatly outperforming vanilla attention-based decoding and achieving BLEU results close to that of offline systems.

The experimental results on Must-C En-De are shown in Table 2, where offline and streaming respectively denote offline Transformer and streaming Transformer models, while Att and CTC/Att represent attention-based beam search and joint CTC/attention simultaneous translation, respectively. The conclusions we get from Must-C En-De are similar to that of the Fisher-CallHome Spanish tasks: 1. The ASR initialization works for both offline and simultaneous ST systems; 2. The BLEU and latency of the simultaneous ST system increase with a larger block size being used; 3. With our proposed cross-lingual encoding method, the simultaneous ST model can further achieve significant improvement with our designed joint CTC/attention simultaneous translation. Furthermore, with our joint CTC/attention simultaneous translation, our simultaneous ST model greatly outperforms the wait-k (k=5 here) prefix-to-prefix model [11]. It should be noted that the wait-k model also uses ASR pre-training initialization method.

4.4. Streaming SLU results

We compare our proposed streaming SLU systems with the offline SLU systems and conduct ablation studies to verify the effectiveness of the ASR-based intermediate loss regularization. The results are shown in Table 3, it should be noted that we predict the intent along with the ASR transcripts [6]. The results show that the streaming SLU system can achieve classification performance close to that of the offline systems, which proves that predicting the intent based on previous chunks and then correcting it using beam search with future chunks works for streaming SLU tasks. In addition, after using the ASR-based intermediate loss, further improvement is achieved, which proves that it is beneficial to let the encoder first learn to distinguish tokens before trying to understand intents.

We also conduct experiments on the SLURP corpus, and the results are shown in Table 4. We can see that the gap between the offline system and the streaming system decreases with the block size increases, although this comes at the cost of increased endpoint latency. Finally, with the help of ASR-based intermediate loss, we achieve 78.3% classification accuracy which is close to that of the offline systems within an acceptable latency.

5. Conclusions

In this paper, we take the first step on applying the blockwise streaming Transformer developed for the ASR task to streaming SLU and simultaneous ST tasks. To further improve the classification performance of the streaming SLU systems, we design an ASR-based intermediate loss, which encourages the encoder to first distinguish tokens before trying to understand intents. As for the simultaneous ST task, we propose a cross-lingual encoding method by injecting a CTC objective between encoder outputs and target translations. In addition, the CTC is also employed to refine the search space and eliminate irregular alignments with the CTC prefix score, achieving joint CTC/attention simultaneous translation. Experimental results show that the blockwise streaming Transformer yields promising results on SLU and ST tasks, especially with our ASR-based intermediate loss or joint CTC/attention simultaneous translation.
6. References

[1] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters, “From audio to semantics: Approaches to end-to-end spoken language understanding,” in SLT, 2018, pp. 720–726.

[2] N. Arivazhagan, C. Cherry, I. Te, W. Macherey, P. Baljekar, and G. Foster, “Re-translation strategies for long form, simultaneous, spoken language translation,” in ICASSP, 2020, pp. 7919–7923.

[3] H.-K. J. Kao, Z. Tüske, S. Thomas, Y. Huang, K. Audhkhasi, B. Kingsbury, G. Kurata, Z. Kons, R. Hoory, and L. Lastras, “End-to-End Spoken Language Understanding Without Full Transcripts,” in Proc. Interspeech 2020, pp. 906–910.

[4] N. Potdar, A. R. Avila, C. Xing, D. Wang, Y. Cao, and X. Chen, “A streaming end-to-end framework for spoken language understanding,” in Proc. IJCAI, 2021.

[5] Y. Qian, R. Ubale, P. Lange, K. Evanini, V. Ramanarayanan, and F. Soong, “Spoken language understanding of human-machine conversations for language learning applications,” Journal of Signal Processing Systems, vol. 92, 08 2020.

[6] S. Arora, S. Dalmia, P. Denisov, X. Chang, Y. Ueda, Y. Peng, Y. Zhang, S. Kumar, K. Ganesan, B. Yan et al., “Espnet-shu: Advancing spoken language understanding through espnet,” arXiv preprint arXiv:2111.14706, 2021.

[7] Y. Liu, J. Zhang, H. Xiong, L. Zhou, Z. He, H. Wu, H. Wang, and C. Zong, “Synchronous speech recognition and speech-to-text translation with interactive decoding,” in AAAI, 2020, pp. 8417–8424.

[8] H. Inaguma, S. Dalmia, B. Yan, and S. Watanabe, “Fastmd: Fast multi-decoder end-to-end speech translation with non-autoregressive hidden intermediates,” in ASRU, 2021, pp. 922–929.

[9] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in ICML, 2014, pp. 1764–1772.

[10] Y. Cao, N. Potdar, and A. R. Avila, “Sequential End-to-End Intent and Slot Label Classification and Localization,” in Proc. Interspeech 2021, 2021, pp. 1229–1233.

[11] X. Ma, J. M. Pino, and P. Koehn, “Simult to simulst: Adapting simultaneous text translation to end-to-end simultaneous speech translation,” in AACL/IJCNLP, 2020, pp. 582–587.

[12] X. Ma, Y. Wang, M. J. Dousti, P. Koehn, and J. M. Pino, “Streaming simultaneous speech translation with augmented memory transformer,” in ICASSP, 2021, pp. 7523–7527.

[13] M. Ma, L. Huang, H. Xiong, R. Zheng, K. Liu, B. Zheng, C. Zhang, Z. He, H. Liu, X. Li, H. Wu, and H. Wang, “STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework,” in Proc. ACL, 2020, pp. 3025–3036.

[14] M. A. Zaidi, B. Lee, N. K. Lakumarapu, S. Kim, and C. Kim, “Decision attentive regularization to improve simultaneous speech translation systems,” CoRR, vol. abs/2110.15729, 2021.

[15] X. Ma, J. M. Pino, J. Cross, L. Puzon, and J. Gu, “Monotonic multthead attention,” in ICLR, 2020.

[16] C. Chiu and C. Raffel, “Monotonic chunkwise attention,” in ICLR (Poster), 2018.

[17] J. Chen, M. Ma, R. Zheng, and L. Huang, “Direct simultaneous speech-to-text translation assisted by synchronized streaming ASR,” in ACL/IJCNLP (Findings), vol. ACL/IJCNLP, 2021, pp. 4618–4624.

[18] E. Tsunoo, Y. Kashiwagi, and S. Watanabe, “Streaming transformer asr with blockwise synchronous beam search,” in SLT, 2021, pp. 22–29.

[19] C. Yi, S. Zhou, and B. Xu, “Efficiently fusing pretrained acoustic and linguistic encoders for low-resource speech recognition,” IEEE Signal Processing Letters, vol. 28, pp. 788–792, 2021.

[20] J. Lee and S. Watanabe, “Intermediate loss regularization for ctet-based speech recognition,” in ICASSP, 2021, pp. 6224–6228.