Large-Scale Streaming End-to-End Speech Translation with Neural Transducers

Jian Xue[*1], *Peidong Wang*[*1], *Jinyu Li*[1], *Matt Post*[2], *Yashesh Gaur*[1]

[1]Microsoft Speech Group
[2]Microsoft Translator Group

{jiaxue, peidongwang, jinyli, mattpost, yagaur}@microsoft.com

Abstract

Neural transducers have been widely used in automatic speech recognition (ASR). In this paper, we introduce it to streaming end-to-end speech translation (ST), which aims to convert audio signals to texts in other languages directly. Compared with cascaded ST that performs ASR followed by text-based machine translation (MT), the proposed Transformer transducer (TT)-based ST model drastically reduces inference latency, exploits speech information, and avoids error propagation from ASR to MT. To improve the modeling capacity, we propose attention pooling for the joint network in TT. In addition, we extend TT-based ST to multilingual ST, which generates texts of multiple languages at the same time. Experimental results on a large-scale 50 thousand (K) hours pseudo-labeled training set show that TT-based ST not only significantly reduces inference time but also outperforms non-streaming cascaded ST for English-German translation.

Index Terms: speech translation, streaming, end-to-end, neural transducers, attention pooling

1. Introduction

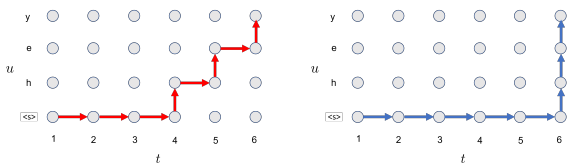
Speech translation (ST) aims to convert speech signals to texts in other languages. Conventionally, it is formulated as a two-step cascaded task, automatic speech recognition (ASR) followed by text-based machine translation (MT) [1, 2, 3]. Such cascaded systems typically suffer from the following issues. First, errors in ASR may propagate to MT. Second, since the intermediate representation is text, cascaded systems cannot fully leverage speech information (e.g., prosody) for translation. Finally, the MT module cannot start until the ASR module has (partially) finished, resulting in long inference latency. Recently, end-to-end (E2E) ST (i.e., direct ST), which directly maps audio features to texts, has become more and more popular [4, 5]. In [6], the authors propose to use attention-based E2E encoder-decoder models (AED) [7, 8] on a small French-English synthetic corpus. In [9], a similar model structure is applied to the Fisher Callhome Spanish-English task and outperforms the cascaded method on the Fisher test set. AED-based models were also used in [10] for a large-scale E2E ST task. However, AED models are usually operated in an offline mode which cannot start decoding until the full utterance is observed.

E2E ST and ASR are similar in that they are both sequence-to-sequence mappings. Many model architectures can thus be shared, especially between ST using monotonic alignments [11] and ASR. To enable more effective communication between users, streaming (i.e., simultaneous) models are topics of investigation in both areas. Monotonic chunkwise attention (MoChA) [12] was used in both MT and ASR. The MT version was extended to monotonic infinite lookback attention (MILk) [13] and monotonic multi-head attention [14, 15], and the ASR version was improved by multitask learning [16] and minimum latency training strategies [17]. Another streaming model architecture is the Neural transducer [18, 19, 20, 21], which outperforms MoChA and has emerged to be the state-of-the-art (SOTA) streaming E2E model in ASR [22], but has been less investigated in ST. Recently, Liu *et al.* proposed cross attention augmented transducer (CAAT) for ST [23]. It uses Transformers in the joint network to combine encoder and prediction network outputs. Due to the use of Transformers and multi-step decision for memory footprint reduction, the latency of CAAT is large. In addition, to train a CAAT, complicated regularization terms and extensive hyper-parameter tuning are required.

In this paper, to leverage the success of the SOTA streaming technology in ASR, we propose to use neural transducers, specifically a low-latency and low-computational-cost Transformer transducer (TT) [24] for streaming E2E ST. To improve the representation fusion ability of the joint networks in TT, we propose *attention pooling*. In addition, we extend TT for multilingual ST. The TT models are trained on a 50K-hour pseudo-labeled ST data set, which is generated by feeding the reference texts in ASR corpus to a MT model [25, 26, 27]. We do not use any human-labeled paired ST data in this work. Experimental results on the Microsoft speech language translation (MSLT) corpus [28] demonstrate that the proposed method not only achieves good bilingual evaluation understudy (BLEU) scores but also significantly reduces inference latency. The remainder of this paper is organized as follows. We describe the proposed methods and model structures in Section 2. The experimental setup and evaluation results are shown in Section 3 and 4. We conclude this paper in Section 5.

2. System Description

2.1. From Wait-k To Neural Transducers



(a) *wait-3* (b) *wait-∞*

Figure 1: *Illustration of the decoding graphs of wait-k. For a given k, the read-write path is deterministic.*

Figure 1 shows the decoding graphs of the commonly adopted wait-k algorithm [29] for streaming ST, where $t \in$

* Equal Contribution

$[1, T]$ and $u \in [1, U]$ denote the time steps for encoder output and output labels (i.e., read and write operations), respectively. As indicated by the names, wait-3 in Figure 1a waits for 3 read operations to start writing, whereas wait-∞ in Figure 1b can access the whole sentence.

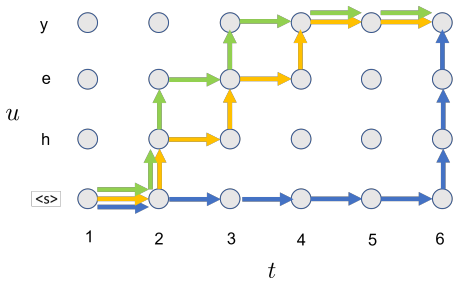


Figure 2: *Illustration of three possible decoding graphs of a neural transducer. The path is chosen dynamically.*

Instead of using hard-coded wait steps and a fixed read-write policy in wait-k, neural transducers, whose decoding graphs are depicted in Figure 2, make read and write decisions in a data-driven fashion. During training, a neural transducer considers all possible alignments between encoder output and labels. At test time, it generates the most likely paths adaptively based on the input features. As shown in Figure 2, if there is no significant word reordering, the neural transducer may follow the orange path or a different green path. If there is a significant word reordering at the end of the utterance, it can use the blue decoding path corresponding to wait-∞.

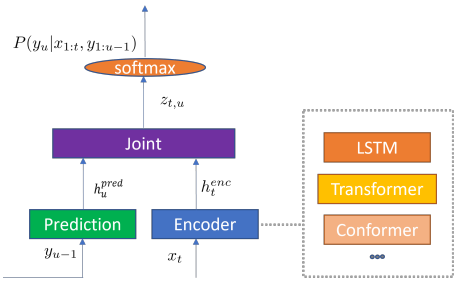


Figure 3: *Illustration of neural transducers.*

The model structure of neural transducers is shown in Figure 3. It has three components: an encoder network, a prediction network, and a joint network. The encoder takes d_x-dimension audio features $\mathbf{x}_t \in \mathbb{R}^{d_x}$ as input and generates d_e-dimension hidden representations $\mathbf{h}_t^{enc} \in \mathbb{R}^{d_e}$. The prediction network uses the embedding of non-blank output token $\mathbf{y}_{u-1} \in \mathbb{R}^1$ at time $u-1$ and predicts hidden representation $\mathbf{h}_u^{pred} \in \mathbb{R}^{d_p}$ for step u. As for the joint network, it combines $\mathbf{h}_t^{enc}$ and $\mathbf{h}_u^{pred}$ to a $T \times U$ tensor, whose element at t and u is denoted by the vector $\mathbf{z}_{t,u} \in \mathbb{R}^{d_z}$. After a softmax operation, the model generates probability $P(\mathbf{y}_u \in \mathbf{Y} \cup \varnothing | \mathbf{x}_{1:t}, \mathbf{y}_{1:u-1})$, where $\mathbf{Y}$ is the vocabulary list and $\varnothing$ denotes the *blank* output (i.e., output nothing). Note that for notation simplicity, we ignore batch size and use the same time resolution for $\mathbf{x}_t$ and $\mathbf{h}_t^{enc}$ in this paper. Neural transducers can use different types of networks as encoders, such as long short-term memory (LSTM) recurrent neural networks (RNNs) in RNN-T models [30] and Transformers in TT models [31, 32].

2.2. Streaming TT Model

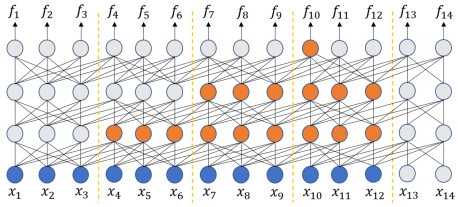


Figure 4: *Illustration of the reception field of a streaming TT at position f_{10} with chunk size 3 and the number of left chunks 1.*

We apply TT in this work since it usually obtains better performance than RNN-T in ASR tasks [24, 31, 32]. Each Transformer block in the encoder is constructed from a multi-head self-attention layer followed by a feedforward layer. In order for TT to work in a streaming mode with low latency and low computational cost, we apply the attention mask proposed in [24]. The attention mask can be the same for different layers. At each layer l, we divide the input $\mathbf{x}_{1:T}^l$ into chunks $\mathbf{c}_{1:S}^l$ along time with chunk size U, where $\mathbf{c}_s^l = \mathbf{x}_{s \times U:(s+1) \times U-1}^l$. At time step t, $\mathbf{x}_t^l$ can only see the frames inside its own chunk $\mathbf{c}_{t/U+1}^l$ and a fixed number B of left chunks $\mathbf{c}_{max(1,t/U+1-B):t/U}^l$. Figure 4 shows an example of the reception field for a three-layer Transformer model with chunk size $U = 3$ and the number of left chunks $B = 1$ at output position f_{10}. Note that since the features cannot access the frames ahead its own chunk, as shown by the right-most blue circle in the first input layer, the number of look-ahead frames is kept $U - 1 = 2$. Moreover, the left reception field increases linearly with the number of layers, enabling the model to use a long history information for a better performance.

2.3. Attention Pooling for Joint Networks

The joint network in a conventional neural transducer combines the output representations of encoder and prediction network with simple linear layers:

$$\mathbf{z}_{t,u} = W^{out} f(W_e^{joint} \mathbf{h}_t^{enc} + W_p^{joint} \mathbf{h}_u^{pred}) \quad (1)$$

where the two sources of output are multiplied with $W_e^{joint} \in \mathbb{R}^{d_e \times d_j}$ and $W_p^{joint} \in \mathbb{R}^{d_p \times d_j}$, to map the feature vectors to d_j-dimension, respectively. f denotes a non-linear function, which is typically $\tanh$ or $relu$. Finally, the feature vector is converted to the output dimension using $W^{out} \in \mathbb{R}^{d_j \times d_z}$.

Recent study in ASR shows that the representation fusion ability of such joint networks can be improved by a bilinear pooling approach [33]. In this paper, we propose attention pooling, which could adapt the pooling weights according to the input using an attention-like weighting mechanism. Different from ASR, ST needs to consider not only the current output probability $P(\mathbf{y}_u \in \mathbf{Y} \cup \varnothing | \mathbf{x}_{1:t}, \mathbf{y}_{1:u-1})$ but also whether writing a non-*blank* token at a future step is better. The adaptive attention weights in attention pooling may work as an additional type of feature to help ST models to make more appropriate decisions. Note that we keep the time and space complexity of attention pooling to be linear so that it consumes less computation resources during inference. The proposed attention pooling is defined as Equation (2) to (5) below:

$$\mathbf{z}_{t,u} = W^{out} f(\hat{\mathbf{h}}_{t,u}^{joint} + W_e^{joint} \mathbf{h}_t^{enc} + W_p^{joint} \mathbf{h}_u^{pred}) \quad (2)$$

$$\hat{\mathbf{h}}_{t,u}^{joint} = W^{proj}(v_t^{enc} \odot v_u^{pred}) \quad (3)$$

$$v_t^{enc} = f((softmax(W^e \mathbf{h}_t^{enc}) \cdot \mathbf{h}_t^{enc})) \quad (4)$$

$$v_u^{pred} = f((softmax(W^p \mathbf{h}_u^{pred}) \cdot \mathbf{h}_u^{pred})) \quad (5)$$

where $\hat{\mathbf{h}}_{t,u}^{joint} \in \mathbb{R}^{d_j}$ is the pooling term at time steps t and u, $W^{proj} \in \mathbb{R}^{1 \times d_j}$ maps the 1-dimension feature to d_j-dimension, $v_t^{enc} \in \mathbb{R}^1$ and $v_u^{pred} \in \mathbb{R}^1$ denotes the contribution of encoder and prediction network to the pooling term, $\odot$ denotes Hadamard product, $W^e \in \mathbb{R}^{d_e \times d_e}$ and $W^p \in \mathbb{R}^{d_p \times d_p}$ are used to calculate the attention weights, and $\cdot$ denotes tensor-dot operation.

We also design a stronger qkv attention pooling method that uses separate weights for query, key, and value. It can be expressed by replacing Equation (4) and (5) with Equation (6) and (7), respectively:

$$v_t^{enc} = f((softmax(W_q^e \mathbf{h}_t^{enc} \odot W_k^e \mathbf{h}_t^{enc}) \cdot (W_v^e \mathbf{h}_t^{enc})) \quad (6)$$

$$v_u^{pred} = f((softmax(W_q^p \mathbf{h}_u^{pred} \odot W_k^p \mathbf{h}_u^{pred}) \cdot (W_v^p \mathbf{h}_u^{pred}))) \quad (7)$$

where $W_q^e \in \mathbb{R}^{d_e \times d_e}$, $W_k^e \in \mathbb{R}^{d_e \times d_e}$, and $W_v^e \in \mathbb{R}^{d_e \times d_e}$ are the linear layers for query, key, and value for encoder features, and $W_q^p \in \mathbb{R}^{d_p \times d_p}$, $W_k^p \in \mathbb{R}^{d_p \times d_p}$, and $W_v^p \in \mathbb{R}^{d_p \times d_p}$ are the corresponding weights for the prediction network. Note that we use Hadamard product and tensor-dot to avoid quadratic time and space complexity.

2.4. Multilingual ST with TT

ST supporting a single language pair such as English-Chinese (EN-ZH) are often referred to as bilingual ST. It is inefficient to build a separate bilingual ST model for every language pair in the world. In addition, running multiple bilingual ST models simultaneously requires a lot of memory and computation resources. In this work, we propose to apply TT to multilingual ST by sharing the encoder and using separate prediction and joint networks for different target languages. Since the encoder (64M of parameters in our experiments) is much larger than joint and prediction networks (24M combined), the size of such a multilingual ST model is comparable to a bilingual model.

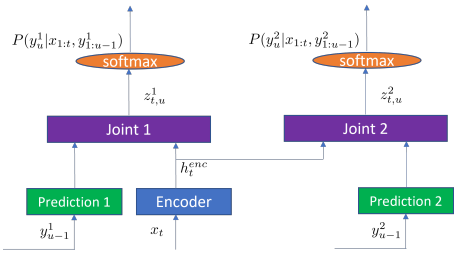


Figure 5: *Illustration of TT for multilingual ST.*

Figure 5 shows the one-to-many multilingual ST model using TT. Shared encoder output $\mathbf{h}_t^{enc}$ is fed to multiple prediction and joint networks. To train the multilingual ST model, we alternate the training data for different batches, e.g., one batch using EN-ZH data and another English-German (EN-DE) (i.e., data mixing ratio is 50%-50%).

3. Experimental Setup

We use 50 thousand (K) hours of Microsoft internal ASR data as the training set. All the data are anonymized with personally identifiable information removed. The original transcriptions are in English, and we use Microsoft cognitive translation service to translate them into Chinese and German. We do not use any human-labeled paired ST data. For evaluation, we use the publicly available MSLT_v1.0 for EN-DE and MSLT_v1.1 for EN-ZH [28]. [1]

3.1. Cascaded Method

We use the cascaded method as the baseline for our experiments. The ASR module is a streaming TT model described in Section 2.2. It is trained using the above 50K-hour English audio and the corresponding English transcriptions. The encoder consists of 18 Transformer blocks, each containing 320 hidden nodes, 8 attention heads, and 2048 feedforward nodes. As for the prediction network, we use 2 LSTM layers. Each LSTM layer has 1024 hidden nodes and the embedding dimension is also 1024. The joint network is a simple feedforward layer containing 512 nodes. For English transcriptions, the vocabulary size is 4K. The model has 88M parameters in total. The input to the model is 80-dimension log-Mel filter-bank features with 25ms windows and 10ms shift, extracted from mixed band training data [34]. Before the input is fed to the Transformer blocks, it is filtered and down-sampled by a factor of 4 (i.e., the resulting sampling rate is 40ms) using two convolutional layers. The chunk size of the streaming mask for the Transformer blocks is 4. The output of the ASR module is used by a non-streaming text-based MT module to get the translation results. The MT model consists of a 6-layer Transformer encoder and a 2-layer RNN decoder. Each Transformer layer has a feedforward network of size 2048 and 8 attention heads. The embedding size is 512. The MT model has 67M parameters.

3.2. Streaming E2E ST Models

The streaming E2E ST models are trained using the same 50K-hour English audio, but with the corresponding translated labels generated by the MT model. The ST models have the same architecture as that of the ASR module in the above cascaded system, except that the output dimensions are changed according to vocabulary sizes. The vocabulary size of EN-ZH is 11K, whereas that of EN-DE is 4K. In addition to chunk size 4, which has 160ms look-ahead and is denoted as TT-160ms, we conduct experiments using a chunk size of 80, whose look-ahead is 3.2s and is thus denoted as TT-3.2s.

3.3. ASR Encoder Initialization and ASR Multi-Task Learning

In addition to pseudo-labeling, we investigate ASR encoder initialization and ASR multi-task learning for ST. For ASR encoder initialization, we use the ASR module from the cascaded method to initialize the encoder and randomly initialize the prediction and joint networks. Then we fine-tune the whole model for the EN-ZH task. As for ASR multi-task learning, we adopt the multilingual ST model described in Section 2.4, but with English and Chinese as the output languages.

3.4. Attention Pooling for Joint Networks

Each new weight matrix introduced by attention pooling is implemented as a linear layer. The open neural network exchange (ONNX) conversion procedures are modified accordingly.

[1] We are not allowed to use other public data sets due to license restrictions.

3.5. ONNX Conversion and Model Compression

After getting the checkpoints of E2E ST models, we convert them to ONNX format, compress each component, and evaluate the compressed models. The weights in the encoders are compressed to uint8, and the RNN and feedforward layers in the prediction and joint networks are compressed using neural-network unified preprocessing heterogeneous architecture (NUPHAR).

3.6. Latency Measurement

We use average proportion (AP), average lagging (AL), and differentiable average lagging (DAL) proposed in [35] to measure the inference latencies of our ST systems. Note that different from [35], our results are generated after the models are converted to ONNX format and compressed.

4. Evaluation Results

4.1. Transformer Transducer for Speech Translation

The first parts of Table 1 and 2 contain the BLEU scores and latencies of the cascaded ST models and TT-based E2E ST models. First, comparing TT-3.2s and TT-160ms, TT-3.2s outperforms TT-160ms in BLEU scores on both EN-ZH and EN-DE, at the cost of a significant latency increase. The differences between TT-3.2s and TT-160ms in BLEU scores are small: 0.7 on EN-ZH and 1.3 on EN-DE, indicating that TT can maintain a good translation quality when working in streaming mode. The AL of TT-3.2s is about 2.2s, shorter than the look-ahead time. The reason is that except for the first output token, TT-3.2s does not have to wait for the whole 3.2s to generate an output. On the contrary, the AL of TT-160ms is 841ms, longer than 160ms. This shows that that TT-160ms requires multiple frames to handle word reordering. Second, comparing the BLEU scores of cascaded models and TT-160ms, we observe that on EN-ZH, there is still a gap. However, on the EN-DE test set, TT-160ms outperforms the cascaded model. Note that TT-160ms is a streaming model with a very small latency, whereas the cascaded model is a non-streaming model and is trained using additional text-text MT data. We also conduct experiments on a non-streaming AED E2E model, but its performance is not as good as TT-160ms. Finally, we mainly use pseudo-labeling to deal with the data scarcity problem in this study. To exploit the 50K-hour ASR training data, we investigate ASR encoder initialization and ASR multi-task learning as described in Section 3.3. As shown in Table 1, these two methods do not help in our experiments, possibly because our ST models are trained with a large amount of training data.

4.2. Attention Pooling for Joint Networks

Table 1 contains the comparison between TT-160ms and different pooling methods for joint networks. Bilinear pooling does not improve the performance of TT-160ms in this study. The reason may be that it lacks the ability to adapt the pooling weights according to the input, which is important in ST since the output of the prediction network is in a different language and is not monotonic w.r.t. the audio features. The attention pooling methods proposed in this paper show consistent BLEU score improvements over TT-160ms. The latencies are also very close to those of TT-160ms. Note that each input frame is 10ms and the encoder has a subsampling factor of 4. The attention pooling methods are thus at most 1-2 steps slower at the encoder output, and the slightly higher latency of simple attention pooling over qkv attention pooling can be negligible. Since the simple attention pooling method obtains a larger BLEU score improvement per additional parameter, which is calculated as $\Delta BLEU/\Delta \#params$, we use it for the evaluation on EN-DE. As shown in Table 2, attention pooling obtains a consistent BLEU score improvement over TT-160ms.

Table 1: *Comparisons of BLEU scores and latencies on EN-ZH of MSLT_v1.1_test. The numbers following the pooling methods denote the number of additional parameters being introduced. The AL and DAL values are in milliseconds (ms).*

methods	BLEU ↑	AP ↓	AL ↓	DAL ↓
cascaded	40.0	1	∞	∞
TT-3.2s	35.6	0.74	2151	1886
TT-160ms	34.9	0.61	841	834
ASR encoder init	34.7	0.61	841	834
ASR multi-task learning	34.7	0.61	841	834
bilinear (+2K)	34.5	0.61	862	859
attention (+1.3M)	35.1	0.62	910	910
qkv attention (+3.9M)	35.3	0.62	875	877
multilingual EN-ZH output	34.8	0.61	841	834

Table 2: *Comparisons of BLEU scores and latencies on EN-DE of MSLT_v1.0_test.*

methods	BLEU ↑	AP ↓	AL ↓	DAL ↓
cascaded	29.3	1	∞	∞
TT-3.2s	30.7	0.74	2152	1890
TT-160ms	29.4	0.61	828	828
attention (+1.3M)	29.6	0.61	871	869
multilingual EN-DE output	29.2	0.61	828	828

4.3. Multilingual ST with TT

The last lines in Table 1 and Table 2 correspond to the EN-ZH output and EN-DE output of the TT-based streaming E2E multilingual ST. Note that although the results are shown in different tables, they are generated simultaneously. The BLEU scores of multilingual ST are slightly worse than those of the bilingual TT-160ms models, the differences are 0.1 for EN-ZH and 0.2 for EN-DE, respectively. In addition to good BLEU scores, multilingual ST greatly reduces the model size and computation burden since it shares a single encoder for multiple languages.

5. Conclusions

We propose neural transducers for large-scale streaming E2E ST. To improve the performance of TT for ST, we propose attention pooling for joint networks. Moreover, we extend TT to multilingual ST by sharing the encoder. Experimental results on the EN-ZH and EN-DE test sets of MSLT show that the proposed TT-based streaming E2E ST models achieve high-quality translation performance with low inference latency. More specifically, the proposed streaming E2E ST system outperforms a non-streaming cascaded system on EN-DE.

6. Acknowledgement

We would like to thank Drs. Long Zhou, Yu Wu, and Shujie Liu at Microsoft Research Asia for valuable suggestions.

7. References

[1] H. Ney, "Speech translation: Coupling of recognition and translation," in *Proceedings of ICASSP*, 1999, pp. 517–520.

[2] E. Matusov, S. Kanthak, and H. Ney, "On the integration of speech recognition and statistical machine translation," in *European Conference on Speech Communicaton and Technology*, 2005.

[3] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, "Improved speech-to-text translation with the fisher and callhome spanish-english speech translation corpus," in *Proceedings of IWSLT*, 2013.

[4] L. C. Vila, C. Escolano, J. A. Fonollosa, and M. R. Costa-Jussa, "End-to-end speech translation with the transformer." in *Proceedings of Interspeech*, 2018, pp. 60–63.

[5] M. Sperber and M. Paulik, "Speech translation and the end-to-end promise: Taking stock of where we are," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7409–7421.

[6] A. Berard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," in *NIPS Workshop on End-to-end Learning for Speech and Audio Processing*, 2016.

[7] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[8] P. Wang, T. N. Sainath, and R. J. Weiss, "Multitask training with text data for end-to-end speech recognition," in *Proc. of Interspeech*, 2021, pp. 2566–2570.

[9] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," *Proceedings of Interspeech*, pp. 2625–2629, 2017.

[10] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2018, pp. 6224–6228.

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

[12] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *International Conference on Learning Representations*, 2018.

[13] N. Arivazhagan, C. Cherry, W. Macherey, C.-C. Chiu, S. Yavuz, R. Pang, W. Li, and C. Raffel, "Monotonic infinite lookback attention for simultaneous machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1313–1323.

[14] X. Ma, J. M. Pino, J. Cross, L. Puzon, and J. Gu, "Monotonic multihead attention," in *Proceedings of International Conference on Learning Representations*, 2019.

[15] X. Ma, Y. Wang, M. J. Dousti, P. Koehn, and J. Pino, "Streaming simultaneous speech translation with augmented memory transformer," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021, pp. 7523–7527.

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

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

[18] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proceedings of Interspeech*, 2017, pp. 939–943.

[19] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S.-Y. Chang, W. Li, R. Alvarez, Z. Chen, and et al, "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *Proceedings of ICASSP*, 2020, pp. 6059–6003.

[20] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao, and et al, "Developing rnnt models surpassing high-performance hybrid models with customization capability," in *Proceedings of Interspeech*, 2020, pp. 3590–3594.

[21] G. Saon, Z. Tüske, D. Bolanos, and B. Kingsbury, "Advancing rnn transducer technology for speech recognition," in *Proceedings of ICASSP*, 2021, pp. 5654–5658.

[22] J. Li, "Recent advances in end-to-end automatic speech recognition," *arXiv preprint arXiv:2111.01690*, 2021.

[23] D. Liu, M. Du, X. Li, Y. Li, and E. Chen, "Cross attention augmented transducer networks for simultaneous translation," in *Proceedings of EMNLP*, 2021, pp. 39–55.

[24] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021, pp. 5904–5908.

[25] Y. Liu, H. Xiong, Z. He, J. Zhang, H. Wu, H. Wang, and C. Zong, "End-to-end speech translation with knowledge distillation," *arXiv preprint arXiv:1904.08075*, 2019.

[26] Y. Jia, M. Johnson, W. Macherey, R. J. Weiss, Y. Cao, C.-C. Chiu, N. Ari, S. Laurenzo, and Y. Wu, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 7180–7184.

[27] M. Gaido, M. A. Di Gangi, M. Negri, and M. Turchi, "End-to-end speech-translation with knowledge distillation," in *Proceedings of the International Conference on Spoken Language Translation*, 2020, pp. 80–88.

[28] C. Federmann and W. D. Lewis, "Microsoft speech language translation (MSLT) corpus: The IWSLT 2016 release for english, french and german," in *Proceedings of IWSLT*, 2016.

[29] M. Ma, L. Huang, H. Xiong, R. Zheng, K. Liu, B. Zheng, C. Zhang, Z. He, H. Liu, X. Li *et al.*, "Stacl: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3025–3036.

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

[31] C.-F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le, M. Jain, K. Schubert, C. Fuegen, and M. Seltzer, "Transformer-transducer: End-to-end speech recognition with self-attention," *arXiv preprint arXiv:1910.12977*, 2019.

[32] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 7829–7833.

[33] C. Zhang, B. Li, Z. Lu, T. N. Sainath, and S.-y. Chang, "Improving the fusion of acoustic and text representations in rnn-t," *arXiv preprint arXiv:2201.10240*, 2022.

[34] J. Li, D. Yu, J.-T. Huang, and Y. Gong, "Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM," in *Proceedings of SLT*. IEEE, 2012, pp. 131–136.

[35] X. Ma, M. J. Dousti, C. Wang, J. Gu, and J. Pino, "SIMULEVAL: An evaluation toolkit for simultaneous translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 144–150.