The HW-TSC's Simultaneous Speech Translation System for IWSLT 2022 Evaluation

Minghan Wang[1], Jiaxin Guo[1], Yinglu Li[1], Xiaosong Qiao[1], Yuxia Wang[2], Zongyao Li[1], Chang Su[1], Yimeng Chen[1], Min Zhang[1], Shimin Tao[1], Hao Yang[1], Ying Qin[1]
[1]Huawei Translation Services Center, Beijing, China
[2]The University of Melbourne, Melbourne, Australia
{wangminghan,guojiaxin1,liyinglu,qiaoxiaosong,lizongyao,suchang8,chenyimeng,zhangmin186,taoshimin,yanghao30,qinying}@huawei.com
yuxiaw@student.unimelb.edu.au

Abstract

This paper presents our work in the participation of IWSLT 2022 simultaneous speech translation evaluation. For the track of text-to-text (T2T), we participate in three language pairs and build wait-k based simultaneous MT (SimulMT) model for the task. The model was pretrained on WMT21 news corpora, and was further improved with in-domain fine-tuning and self-training. For the speech-to-text (S2T) track, we designed both cascade and end-to-end form in three language pairs. The cascade system is composed of a chunking-based streaming ASR model and the SimulMT model used in the T2T track. The end-to-end system is a simultaneous speech translation (SimulST) model based on wait-k strategy, which is directly trained on a synthetic corpus produced by translating all texts of ASR corpora into specific target language with an offline MT model. It also contains a heuristic sentence breaking strategy, preventing it from finishing the translation before the the end of the speech. We evaluate our systems on the MUST-C tst-COMMON dataset and show that the end-to-end system is competitive to the cascade one. Meanwhile, we also demonstrate that the SimulMT model can be efficiently optimized by these approaches, resulting in the improvements of 1-2 BLEU points.

1 Introduction

Simultaneous speech/text translation (SimulST/SimulMT) applications are widely demanded in international communication scenarios such as conferences or live streaming.

From the perspective of system architecture, recent works on SimulST can be classified into cascade and end-to-end forms. Cascade systems are often composed of a streaming Automatic Speech Recognition (ASR) module and a steaming text-to-text machine translation module (MT). It might also contains other correction modules. The integration of these modules can be challenging, but the training of each can be beneficial from sufficient data resources. End-to-end approach is also a choice for SimulST, where translations can be directly generated from a unified model with the speech inputs, but bilingual speech translation datasets are still scarce resources.

From the perspective of simultaneous strategy, there is a fixed strategy which is represented by wait-k (Ma et al., 2019) and a flexible strategy such as monotonic attention (Arivazhagan et al., 2019). The fixed strategy is easier to implement but with inferior performance and the flexible one is more robust to the speed of speech but can be non-trivial in the implementation and training. Re-translation is also a strategy proposed recently for SimulMT system, which benefits from pre-trained MT models but often encounters with flicker (Arivazhagan et al., 2020; Sen et al., 2021).

The IWSLT 2022 SimulST shared task (Anastasopoulos et al., 2022) aims to provide a platform for participants to evaluate their approaches on both quality and latency. In this year, there are two sub-tracks, i.e. speech-to-text (S2T) and text-to-text (T2T), and three language directions including En-Zh, En-De and En-Ja in the evaluation. All submitted systems will be evaluated with the SimulEval (Ma et al., 2020a) tool, where BLEU (Papineni et al., 2002) and Average Lagging (AL) (Ma et al., 2020a) are used as metrics for ranking. Meanwhile, systems will be classified into three latency regimes (low, medium, high) with their AL, which are determined differently by the language pairs. The SimulEval formulates the simultaneous translation as a process where an agent should take "READ" or "WRITE" actions to control the progress of translation. A "READ" action allows the agent to get the latest source segments from the server. A "WRITE" action enables the agent to make prediction and send generated tokens back to server for scoring. Participants are required to implement their approaches under this framework.

Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022), pages 247 - 254

Dataset	Number of Utterance	Duration (hrs)
Librispeech	281,241	960.85
MuST-C	340,421	590.67
IWSLT	170,229	254.41
CoVoST	1362,422	1802.52
TEDLIUM3	268,214	453.42

Table 1: Data statistics of our ASR corpora

Language	WMT Bilingual	In-domain Text
En-De	79M	459K
En-Zh	96M	590K
En-Ja	42M	552K

Table 2: Data statistics of our MT corpora

In this paper, we present our work on the participation of all language directions for both S2T and T2T sub-tasks. For the T2T task, we start by modeling with the original wait-k model and optimizing it with in-domain fine-tuning and self-training (Gaido et al., 2020), resulting in large improvements on their performance. We experiment both cascade and end-to-end systems for the S2T task and find that the end-to-end one is quite competitive especially on the latency metric.

2 Method

2.1 Data Preparation & Pre-Processing

ASR Corpora We adopt exactly same data pre-processing pipeline to our offline task submission. Briefly, we combine 5 ASR (LibriSpeech(Panayotov et al., 2015), MuST-C V2 (Cattoni et al., 2021), CoVoST (Wang et al., 2020), TED-LIUM 3 (Hernandez et al., 2018) and IWSLT official dataset) corpora and perform strict cleansing based on absolute frame length (within 50 to 3000), number of tokens (within 1 to 150) and the speed of speech (within $\mu(\tau) \pm 4 \times \sigma(\tau)$, where $\tau = \frac{\text{\# frames}}{\text{\# tokens}}$) for all training utterances. There are basically 1% of noisy samples being filtered out.

MT corpora We follow the pipeline in (Wei et al., 2021) to pre-process the WMT 21 news corpora as well as the in-domain corpora (mixture of MUST-C and IWSLT). Statistics of our MT corpora are shown in Table 2.

2.2 ASR model

We adopt the U2 (Zhang et al., 2020) as the ASR module in our cascade system. U2, a framework that can be applied on standard Transformer (Vaswani et al., 2017) or Conformer (Gulati et al., 2020) architectures, is able to perform both streaming and non-streaming ASR. The major difference between U2 and other offline autoregressive ASR models is that it supports streaming with the help of the dynamic chunk training and decodes with a CTC decoder on the top of the encoder. The dynamic chunk training is achieved by dynamically applying a causal mask with different chunk size at the self-attention layer in the encoder. It is similar to the self-attention of an autoregressive decoder, but allowing the hidden representation to condition on some look-ahead contexts within the chunk. During inference, since the encoder hidden states is monotonically encoded chunk by chunk, the argmax decoding of CTC makes sure that tokens decoded in previous chunks are fixed, which successfully achieves streaming. Besides the CTC decoder, U2 also preserves the standard autoregressive (AR) Transformer decoder, and can be jointly trained with the CTC decoder to improve the stability of training. Originally, the AR decoder can be used to re-score CTC generated texts if prefix beam search is used to propose multiple candidates. However, we don't use the re-scoring in our system.

Since the decoding of arbitrary size of the chunk is learned with the dynamic chunk training, the latency of U2 can be freely determined by the chunk size used in the inference. The chunk size is also directly correlated to the performance, as it defines the volume of look-ahead contexts used in the current chunk.

2.3 Text to Text Model

Our T2T models are used in the T2T track and also as the translation module in the cascade system. It is a standard Transformer model with the wait-k strategy (Ma et al., 2019) for simultaneous decoding. For each language pair, we pre-train the wait-k T2T model on the WMT 2021 news corpora following similar settings as (Wei et al., 2021) to acquire the model $\mathcal{M}_1$. Then, we fine-tune it on the mixture of MuST-C and IWSLT corpora denoted as $\mathcal{C}_{\text{ind}}$, and obtain the domain adapted model $\mathcal{M}_2$. Although the domain transferring contributes some improvements, we find that it is not able to solve a key problem. Since the simultaneous decoding is only conditioned on partially observed context, there is a big gap between the training of offline MT models and SimulMT models, in which the

re-ordered translations from unseen context can be significantly difficult for SimulMT models to learn.

To mitigate this problem, we propose to use self-training (Liu et al., 2021; Kim and Rush, 2016). Firstly, we translate the in-domain corpora $\mathcal{C}_{\text{ind}}$ with $\mathcal{M}_2$ and obtain $\mathcal{C}_{\text{ind}}'$, then, we fine-tune $\mathcal{M}_2$ on the mixture of $\mathcal{C}_{\text{ind}}$ and $\mathcal{C}_{\text{ind}}'$ and obtain $\mathcal{M}_3$. In this way, the self-distilled translations are more monotonic and easier to learn.

2.4 Cascade Speech to Text Model

Algorithm 1 Decoding of Cascade System

Require: ASR, T2T, chunk size,k: $\phi, \mathcal{M}, N_c, k$
  Initialize: Speech buffer $S \leftarrow \{\}$
  Initialize: ASR buffer $A \leftarrow \{\}$
  Initialize: MT buffer $H \leftarrow \{\}$
  Initialize: Frame position $p \leftarrow 0$
  Initialize: MT Finish writing chunk $e \leftarrow$ true
  while w is not </s> **do**
    if $|S| - p < N_c$ and e and not finish reading **then**
      READ next input s
      $S \leftarrow S \cup \{s\}$
    else
      $A \leftarrow \phi(S)$: decode all texts with ASR
      $p \leftarrow |S|$: move frame position
      if $|A| - |H| \geq k$ **then**
        decode with MT: $w \leftarrow \mathcal{M}(A)$
        $H \leftarrow H \cup \{w\}$
        $e \leftarrow (|A| - |H| < k)$
        WRITE w
      end if
    end if
  end while

Our cascade system is the integration of U2 and wait-k T2T model. When evaluating with SimulEval, U2 makes decisions mainly based on whether the input stream can fill a chunk, if not, it directly calls READ, otherwise, it transcribes audio inputs into English texts, and passes the entire sequence to the T2T model. The T2T model takes the output of U2 as inputs, and determines whether to read more based on the length difference between source and target sequence compared to k. Note that since U2 may decode several tokens in the latest chunk at once, we need to distinguish the read action of T2T model and ASR model. More specifically, when tokens decoded in the latest chunk from U2 exceeds the length difference of k for the T2T model, we need to let the T2T model decode for several steps instead of using the read action outputs by T2T model to read more audio frames, this will significantly increase the latency. Therefore, we introduce a flag e, representing whether the T2T model finishes its decoding process for all newly input tokens from current chunk. Algorithm 1 and Figure 1 describes the detailed process.

2.5 End-to-end Speech to Text Model

Besides the cascade system, we also explored the end-to-end (E2E) system. A key disadvantage to train an E2E system comes from the lack of large scale speech translation corpora. Therefore, we use the pre-trained MT model (trained on WMT21 News corpora) to create the knowledge distilled data (Kim and Rush, 2016) by translating all ASR corpora into required language, which significantly increases the scale of the training set.

There are two reasons that we use an offline MT model instead of our T2T model to generate the KD data. 1) the T2T model has lower performance compared to the offline model which may further limit the performance upper bound of the student model. 2) Decoding with T2T model is quite slower than the offline MT model.

For the E2E S2T model, we use the Conv-Transformer (Inaguma et al., 2020) with wait-k strategy of different k for each language. More specifically, we adopt similar configurations in (Ma et al., 2020b), where a pre-decision module is used to handle the large length gap between speech frames and target sentence, so that the wait-k algorithm can work properly with enough source information. Here we use the fixed pre-decision policy by pooling frames into a summarized feature vector for the wait-k decision every fixed number of frames (7 frames for all three models in our experiments).

During the evaluation with SimulEval, we found that E2E S2T model can easily predict the "</s>" when there is a silence interval in the speech. Although fed with more source inputs or applied with EOS penalty, the model is still incapable of translating samples into multiple sentences.

We suspect that the model is only trained on properly segmented utterances containing scarce samples with more than one sentence, but evaluated on samples with multiple sentences. This often causes the agent to send an incomplete translation to the server. To this end, we design a simple but efficient sentence breaking strategy to prevent the

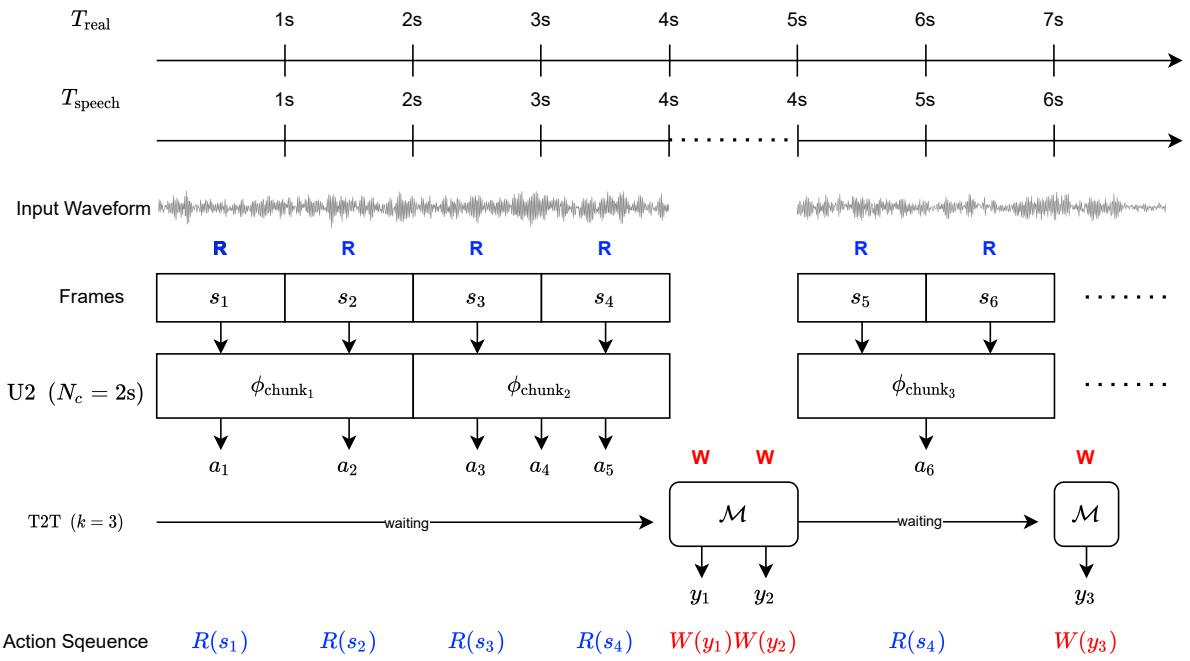

Figure 1: This figure presents an example of decoding with our cascade system, in which the the chunk size of U2 is set equivalent to 2s, the k for the wait-k T2T is set to 3. We plot the timeline of the real wall time and the speech time for a more cleared description. To present the collaboration of two models, we assume that decoding with U2 needs no time but decoding with wait-k T2T requires 0.5s per token.

agent from early stopping. In detail, when the decoder predict "</s>" as the next token, we check if the agent finishes reading source inputs. If it does, the "</s>" is the true ending of the speech, otherwise, it will be used as an ending of the sub-sentence, meaning that the "</s>" won't be sent back to the server, and the agent should keep translating until the entire speech is processed. The ending of a sub-sentence will also be used to clean the source input buffer and target context buffer, which means each sub-sentence is translated independently by the agent. We find this approach may in some extent introduce more latency since for each sub-sentence, the agent needs re-wait-k steps to start the generation, however, it is quite helpful to improve the performance on samples that might be mis-segmented with the original approach.

2.6 Domain Controlled Generation

As mentioned in section 2.1, we combine different corpora with different data source to create the united dataset, in which the domain and text style can be various. Directly training the model on the mixture of them can be harmful to the performance since some of these differences can't be easily captured from the speech inputs, so they should be considered as prior knowledge. Therefore, we reuse the strategy from our last year's work (Wang et al., 2021) by providing a domain tag as a known condition to control the generation style. This strategy is used in our E2E S2T model and ASR model. For the S2T model, we add the domain tag as the first token input to the decoder. For the ASR model, since we only use the CTC decoding, domain information needs to be provided at the encoder side. Therefore, we first encode the domain tag with the word embedding layer of the decoder to acquire its representation vector, then, we perform an element-wise sum with the down-sampled input features before feeding to encoder attention layers.

Since the test sets have similar distribution with MuST-C corpora in previous years, we control the model to generate MuST-C alike text by using the domain tag "<MC>" during the inference process.

3 Experiments

We conduct experiments on three types of systems including T2T, cascade S2T and E2E S2T. All systems are evaluated on the MuST-C tst-COMMON dataset for all three languages.

3.1 Setup

We adopt same configuration recipe to our offline submission on the training of the U2 model,

Language	k	Quality BLEU	Latency AL	AL_CA	AP	AP_CA	DAL	DAL_CA
En-De	k=3	24.98	2.66	-	0.66	-	4.14	-
	k=6	31.50	5.58	-	0.78	-	6.53	-
	k=15	33.38	11.12	-	0.93	-	11.87	-
En-Ja	k=6	8.55	1.74	-	0.67	-	5.70	-
	k=10	14.53	6.70	-	0.85	-	8.53	-
	k=14	14.26	9.75	-	0.92	-	10.95	-
En-Zh	k=6	22.53	2.93	-	0.71	-	5.40	-
	k=10	26.45	6.78	-	0.85	-	8.29	-
	k=14	27.54	9.53	-	0.92	-	10.60	-

Table 3: This table shows the results of our T2T models, where AL is computed with number of tokens.

Language	k	Quality BLEU	Latency AL	AL_CA	AP	AP_CA	DAL	DAL_CA
En-De	k=3	18.56	1959.58	2672.29	0.79	1.02	2411.61	3186.99
	k=6	23.90	2608.47	3490.75	0.87	1.18	3067.46	4110.86
	k=15	24.78	4020.55	5116.26	0.96	1.32	4312.52	5582.31
En-Ja	k=6	7.28	2215.07	2555.88	0.80	0.92	2620.34	2852.7
	k=10	12.16	2867.81	3262.79	0.92	1.06	3343.08	3675.45
	k=14	11.57	3365.65	3764.64	0.95	1.09	3811.56	4142.38
En-Zh	k=6	18.59	2119.71	2468.9	0.83	0.95	2603.03	2837.85
	k=10	22.50	2838.8	3207.05	0.92	1.05	3292.46	3573.82
	k=14	23.61	3424.94	3780.95	0.95	1.09	3782.05	4065.2

Table 4: This table shows the results of our cascade S2T models, where AL is computed with milliseconds.

where 80 dimensional Mel-Filter bank features are extracted from raw waveform, and being augmented with speed perturbation (Ko et al., 2015) and spectral augmentation (Park et al., 2019). The model is trained with the hyper-parameters ($n_{(encoder+decoder)\_layers}$ = 12 + 3, n_{heads} = 8, d_{hidden} = 512, d_{FFN} = 2048, $n_{\text{sub sampling}}$=4) for 50 epochs on 8 V100 GPUs. All ASR texts are tokenized with SPM (Kudo and Richardson, 2018) with the vocab size set as 20000.

For the T2T model, we train three models with different k for each language, where k=(3,6,15) for En-De, k=(6,10,14) for En-Zh, k=(6,10,14) for En-Ja. All of them are trained for 40 epochs with similar hyper-parameters ($n_{(encoder+decoder)\_layers}$ = 16 + 4, n_{heads} = 8, d_{hidden} = 512, d_{FFN} = 2048) while pre-training and 10 epochs for fine-tuning and self-training. For En-De and En-Ja, we use SPM for tokenization with vocab size set to 32k, and subword-nmt for En-Zh with vocab size set to 30k. Note that the vocabularies for T2T models are different from that for the ASR model, meaning that the outputs of ASR model in the cascade system need to be re-tokenized for T2T models.

Three S2T models are trained for each language with k=7 for En-De, k=14 for En-Zh and En-Ja. The hyper-parameters are: ($n_{(encoder+decoder)\_layers}$ = 12 + 6, n_{heads} = 8, d_{hidden} = 512, d_{FFN} = 2048) for all models. We train them for 50 epochs on the knowledge distilled dataset.

3.2 Results

T2T Table 3 shows the results of all T2T models, which are evaluated with the SimulEval with the oracle English texts as source inputs. We can see that for all language pairs, a large improvements can be obtained from low latency to medium latency by increasing k from 3 to 6 (En-De) or from 6 to 10 (En-Zh/Ja), but when increasing the latency

Language	k	Quality BLEU	Latency AL	AL_CA	AP	AP_CA	DAL	DAL_CA
En-De	k=7	22.13	2374.54	2831.08	0.86	0.99	2523.52	2990
En-Ja	k=14	12.82	1848.46	2369.75	0.94	1.09	3374.76	3796.14
En-Zh	k=14	20.38	1753.37	2240.23	0.94	1.09	3341.84	3762.65

Table 5: This table shows the results of our end-to-end S2T models, where AL is computed with milliseconds.

from medium to high, the profit is not that significant, demonstrating that the upper bound of wait-k models can be easily reached even with larger k.

Cascade S2T Table 4 presents result of our cascade S2T models, evaluated with the SimulEval by using utterance speech as inputs. Compared with the oracle inputs of T2T model, the performance of cascade S2T models often degrades 2-4 BLEU points when using the same T2T model due to the error propagation comes from the ASR model. We also find that the latency of our cascade systems are quite large although with relatively low k value. This can be explained from the example in Figure 1 where the wait-k model has to wait until the U2 reads 4 times and completes the decoding of chunk 2 (output 3 tokens), since the wait-k model can only decode when the the length difference satisfies the criteria of k. Unfortunately, this eventually increases the delay of y_1 and y_2 when computing the AL.

End-to-end S2T Table 5 are results from our E2E S2T models. Compared with cascade S2T models, the latency of E2E models can be better controlled since the latency offset caused by the collaboration of the ASR and T2T in the cascade system is not necessarily existed in the E2E model. Surprisingly, the performances of E2E models are also competitive to cascade systems, demonstrating that training the model on KD corpora is quite effective.

3.3 Ablation Study

To further explore the effect of fine-tuning and self-training on our T2T models, we present our experimental results on MuST-C tst-COMMON evaluated for the T2T task as described in Table 6. For all language pairs, in-domain fine-tuning brings 2+ BLEU points and self-training brings additional 1+ points.

Approach	En-De	En-Ja	En-Zh
Pre-training	29.21	11.21	23.14
+Fine-tuning	32.05	13.08	25.73
+Self-Training	33.38	14.26	27.54

Table 6: This table presents the improvements coming from applying each strategy during the training of T2T models. We only present results of models with k=15 for En-De, k=14 for En-Ja and En-Zh.

4 Conclusion

In this paper, we report our work in the IWSLT-2022 simultaneous speech translation evaluation. We explored 4 solutions with a cascade and end-to-end system on two sub-tracks and three language directions: 1) We evaluated the method of training a streaming ASR model U2 on the large scale mixed training corpora and inference with the domain controlled generation. 2) We explored the optimization of wait-k T2T models with self-training, and obtained positive results. 3) We tried to build a cascade S2T system by integrating the streaming ASR model with the wait-k T2T model, and compared it with our end-to-end approach. 4) We trained our end-to-end S2T model with knowledge distillation and found it to be competitive to our cascade approach.

In our future works, we will investigate more in terms of simultaneous strategies, efficient using of pretrained models, as well as better training schema with limited ST dataset.

References

Antonios Anastasopoulos, Luisa Bentivogli, Marcely Z. Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Marcello Federico, Christian Federmann, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel M. Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, Juan Pino, Eliz-

abeth Salesky, Jiatong Shi, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alex Waibel, Changhan Wang, and Shinji Watanabe. 2022. FINDINGS OF THE IWSLT 2022 EVALUATION CAMPAIGN. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, Dublin, Ireland. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1313–1323. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George F. Foster. 2020. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation, IWSLT 2020, Online, July 9 - 10, 2020*, pages 220–227. Association for Computational Linguistics.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Must-c: A multilingual corpus for end-to-end speech translation. *Comput. Speech Lang.*, 66:101155.

Marco Gaido, Mattia Antonino Di Gangi, Matteo Negri, and Marco Turchi. 2020. End-to-end speech-translation with knowledge distillation: Fbk@iwslt2020. In *Proceedings of the 17th International Conference on Spoken Language Translation, IWSLT 2020, Online, July 9 - 10, 2020*, pages 80–88. Association for Computational Linguistics.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 5036–5040. ISCA.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia A. Tomashenko, and Yannick Estève. 2018. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer - 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18-22, 2018, Proceedings*, volume 11096 of *Lecture Notes in Computer Science*, pages 198–208. Springer.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. 2020. Espnet-st: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 302–311. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1317–1327. The Association for Computational Linguistics.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 3586–3589. ISCA.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.

Dan Liu, Mengge Du, Xiaoxi Li, Yuchen Hu, and Lirong Dai. 2021. The USTC-NELSLIP systems for simultaneous speech translation task at IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation, IWSLT 2021, Bangkok, Thailand (online), August 5-6, 2021*, pages 30–38. Association for Computational Linguistics.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3025–3036. Association for Computational Linguistics.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Miguel Pino. 2020a. SIMULEVAL: an evaluation toolkit for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 144–150. Association for Computational Linguistics.

Xutai Ma, Juan Miguel Pino, and Philipp Koehn. 2020b. Simulmt to simulst: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7,*

2020, pages 582–587. Association for Computational Linguistics.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 5206–5210. IEEE.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 2613–2617. ISCA.

Sukanta Sen, Ulrich Germann, and Barry Haddow. 2021. The university of edinburgh's submission to the IWSLT21 simultaneous translation task. In *Proceedings of the 18th International Conference on Spoken Language Translation, IWSLT 2021, Bangkok, Thailand (online), August 5-6, 2021*, pages 46–51. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Changhan Wang, Anne Wu, and Juan Pino. 2020. Covost 2: A massively multilingual speech-to-text translation corpus.

Minghan Wang, Yuxia Wang, Chang Su, Jiaxin Guo, Yingtao Zhang, Yujia Liu, Min Zhang, Shimin Tao, Xingshan Zeng, Liangyou Li, Hao Yang, and Ying Qin. 2021. The hw-tsc's offline speech translation systems for IWSLT 2021 evaluation. *CoRR*, abs/2108.03845.

Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang, and Ying Qin. 2021. Hw-tsc's participation in the WMT 2021 news translation shared task. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 225–231. Association for Computational Linguistics.

Binbin Zhang, Di Wu, Zhuoyuan Yao, Xiong Wang, Fan Yu, Chao Yang, Liyong Guo, Yaguang Hu, Lei Xie, and Xin Lei. 2020. Unified streaming and non-streaming two-pass end-to-end model for speech recognition. *CoRR*, abs/2012.05481.