The USTC-NELSLIP Systems for Simultaneous Speech Translation Task at IWSLT 2021

Dan Liu[1,2], Mengge Du[2], Xiaoxi Li[2], Yuchen Hu[1], and Lirong Dai [1]

[1]University of Science and Technology of China, Hefei, China
[2]iFlytek Research, Hefei, China
{*danliu, huyuchen*}@*mail.ustc.edu.cn*
lrdai@ustc.edu.cn
{*danliu, xxli16,mgdou*}@*iflytek.com*

Abstract

This paper describes USTC-NELSLIP's submissions to the IWSLT2021 Simultaneous Speech Translation task. We proposed a novel simultaneous translation model, Cross-Attention Augmented Transducer (CAAT), which significantly outperforms wait-k baseline. We build speech-to-text and text-to-text simultaneous translation systems based on CAAT models and data augmentation.

1 Introduction

This paper describes the submission to IWSLT 2021 Simultaneous Speech Translation task by National Engineering Laboratory for Speech and Language Information Processing (NELSLIP), University of Science and Technology of China, China.

Simultaneous translation, which translates concurrently with the source-language speech, is widely used in international conference, summits and business. Two problems should be solved to build high-quality low-latency simultaneous translation systems, one is the policy of deciding the READ/WRITE action at each step, and the second is how to output a high-quality target sentence based on partial source sentence.

In this work, we build simultaneous translation systems for both text-to-text (T2T) and speech-to-text S2T) task. We propose a novel architecture, Cross-Attention Augmented Transducer (CAAT), which significantly outperforms wait-k (Ma et al., 2019) baseline in both text-to-text and speech-to-text simultaneous translation task. Besides, we adopt a variety of data augmentation methods, which includes back-translation (Edunov et al., 2018), Self-training (Kim and Rush, 2016a) and speech synthesis with Tacotron2 (Shen et al., 2018). Combining all of these and models ensemble we achieved about 10 BLEU (in S2T task) and 5 BLEU (in T2T task) gains compared to the best performance last year.

This paper is arranged as follows. Firstly we start by analyzing the available training data and our augmentation methods in Sec. 2 . Then after detailed description of our methods and model architectures in Sec. 3, we show our performance in Sec. 4.

2 Data

2.1 Statistics and Preprocessing

EN→DE Speech Corpora The speech datasets used in our experiments are shown in Table 1, where MuST-C, Europarl and CoVoST2 are speech translation specific (speech, transcription and translation included), and LibriSpeech, TED-LIUM3 are speech recognition specific (only speech and transcription). After augmented with speed and echo perturbation, we use Kaldi (Povey et al., 2011) to extract 80 dimensional log-mel filter bank features, computed with a $25ms$ window size and a $10ms$ window shift, and specAugment (Park et al., 2019) were performed during training phase.

Corpus	Segments	duration(h)
MuST-C	250.9k	448
Europarl	69.5k	155
CoVoST2	854.4k	1090
LibriSpeech	281.2k	960
TED-LIUM3	268.2k	452

Table 1: Statistics of speech corpora

Text Translation Corpora The bilingual parallel datasets for Englith to German(EN→DE) and English to Japanese (EN→JA) used are shown in Table 2, and the monolingual datasets in English, German and Japanese are shown in Table 3. And we found the Paracrawl dataset in EN→DE task is too big to our model training, we randomly select a subset of 14M sentences from it.

	Corpus	Sentences
EN-DE	MuST-C-v2	229.7k
	Europarl-v10	1828.5k
	Rapid-2019	1531.3k
	WIT3-TED	209.5k
	Commoncrawl	2399.1k
	WikiMatrix-v1	6227.2k
	Wikititles-v2	1382.6k
	Paracrawl	82638.2k
EN-JA	WIT3-TED	225.0k
	JESC-v2	2797.4k
	kftt	440.3k
	WikiMatrix-v1	3896.0k
	Wikititles-v2	706.0k
	Paracrawl	10120.0k

Table 2: Statistics of translation parallel datasets

language	Corpus	Sentences
EN	Europarl-v10	2295.0k
	News-crawl-2019	33600.8k
DE	Europarl-v10	2108.0k
	News-crawl-2020	53674.4k
JA	News-crawl-2019	3446.4k
	News-crawl-2020	10943.3k

Table 3: Statistics of monolingual datasets

For EN→DE task, we directly use SentencePiece (Kudo and Richardson, 2018) to generate a unigram vocabulary of size 32,000 for source and target language jointly. And for EN→JA task, sentences in Japanese are firstly participled by MeCab (Kudo, 2006), and then a unigram vocabulary of size 32,000 is generated for source and target jointly similar to EN→DE task.

During data preprocessing, the bilingual datasets are firstly filtered by length less than 1024 and length ratio of target to source $0.25 < r < 4$. In the second step, with a baseline Transformer model trained with only bilingual data, we filtered the mismatched parallel pairs with log-likelihood from the baseline model, threshold is set to -4.0 for EN→DE task and -5.0 for EN→JA task. At last we keep 27.3 million sentence pairs for EN-DE task and 17.0 sentence pairs for EN→JA task.

2.2 Data Augmentation

For text-to-text machine translation, augmented data from monolingual corpora in source and target language are generated by self-training (He et al., 2019) and back translation (Edunov et al., 2018) respectively. Statistics of the augmented training data are shown in Table 4.

Data	EN→DE	EN→JA
bilingual data	27.3M	17.0M
+back-translation	34.3M	22.0M
+self-training	41.3M	27.0M

Table 4: Augmented training data set for text-to-text translation.

We further extend these two data augmentation methods to speech-to-text translation, detailed as:

1. Self-training: Maybe similar to sequence-level distillation (Kim and Rush, 2016b; Ren et al., 2020; Liu et al., 2019). Transcriptions of all speech datasets (both speech recognition and speech translation specific) are sent to a text translation model to generate text $y^{'}$ in target language, the generated $y^{'}$ with its corresponding speech are directly added to speech translation dataset.

2. speech synthesis: We employ Tacotron2 (Shen et al., 2018) with slightly modified by introducing speaker representations to both encoder and decoder as our text-to-speech (TTS) model architecture, and trained on MuST-C v2 speech corpora to generate filter-bank speech representations. We randomly select 4M sentence pairs from EN→DE text translation corpora and generate audio feature by speech synthesis. The generated filter bank features and their corresponding target language text are used to expand our speech translation dataset.

The expanded training data for speech translation are shown in Table 5.

Dataset	segements	duration(h)
raw S2T dataset	117.4k	1693
+Self-training	289.8k	4799
+Speech synthesis	721.9k	10424

Table 5: Expanded speech translation dataset with self-training and speech synthesis.

3 Methods and Models

3.1 Cross-Attention Augmented Transducer

Let $\mathbf{x}$ and $\mathbf{y}$ denote the source sequence and target sequence.The policy of simultaneous translation is denoted as an action sequence $\mathbf{p} \in \{R, W\}^{|\mathbf{x}|+|\mathbf{y}|}$ where R denotes the READ action and W the WRITE action.Another representation of policy is extending target sequence $\mathbf{y}$ to length $|\mathbf{x}|+|\mathbf{y}|$ with blank symbol ϕ as $\hat{y} \in (\mathbf{v} \cup \{\phi\})^{|\mathbf{x}|+|\mathbf{y}|}$, where $\mathbf{v}$ is the vocabulary of the target language. The mapping from $\mathbf{y}$ to sets of all possible expansion $\hat{y}$ denotes as $H(\mathbf{x}, \mathbf{y})$.

Inspired by RNN-T (Graves, 2012), the loss function for simultaneous translation can be defined as the marginal conditional probability and expectation of latency metric through all possible expanded paths:

$$
\begin{aligned}
&\mathcal{L}(x, y) = \mathcal{L}_{nll}(x, y) + \mathcal{L}_{latency}(x, y) \\
&= -\log \sum_{\hat{y}} p(\hat{y}|x) + \mathbb{E}_{\hat{y}} d(\hat{y}) \\
&= -\log \sum_{\hat{y}} p(\hat{y}|x) + \sum_{\hat{y}} \Pr(\hat{y}|y, x) d(\hat{y})
\end{aligned} \tag{1}
$$

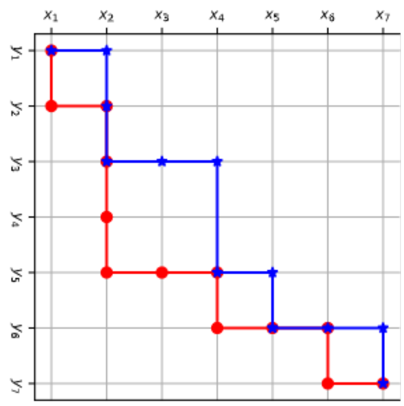


Figure 1: Expanded paths in simultaneous translation

Where $\Pr(\hat{y}|y, x) = \frac{p(\hat{y}|x)}{\sum_{\hat{y}' \in H(x,y)} p(\hat{y}'|x)}$, and $\hat{y} \in H(x, y)$ is a expansion of target sequence $\mathbf{y}$.

However, RNN-T is trained and inferenced based on source-target monotonic constraint, which means it isn't suitable for translation task. And the calculation of marginal probability $\sum_{\hat{y} \in H(x,y)} \Pr(\hat{y}|x)$ is impossible for Attention Encoder-Decoder framework due to deep coupling of source and previous target representation. As shown in Figure 1, the decoder hidden states for the red path $\hat{y}^1$ and the blue path $\hat{y}^2$ is not equal at the intersection $s_2^1 \neq s_2^2$. To solve this, we separate the source attention mechanism from the target history represenation, which is similar to joiner and predictor in RNN-T. The novel architecture can be viewed as a extension version of RNN-T with attention mechanism augmented joiner, and is named as Cross Attention Augmented Transducer (CAAT). Figure 2 is the implementaion of RAAT based on Transformer.

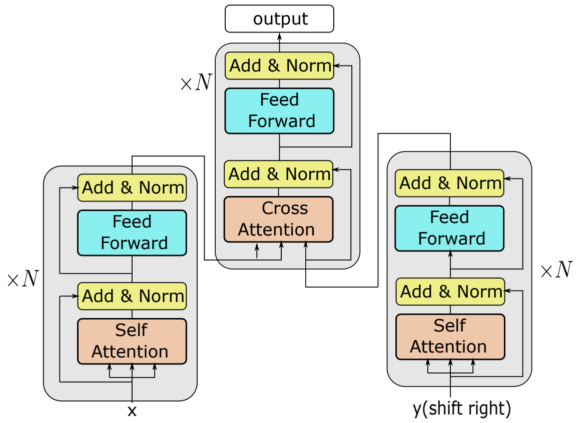


Figure 2: architectures of CAAT based on Transformer

Computation cost of joiner in CAAT is significantly more expensive than that of RNN-T. The complexity of joiner is $\mathcal{O}(|\mathbf{x}| \cdot |\mathbf{y}|)$ during training, which means $\mathcal{O}(|\mathbf{x}|)$ times higher than conventional Transformer. We solve this problem by make decision with decision step size $d > 1$, and reduce the complexity of joiner from $\mathcal{O}(|\mathbf{x}| \cdot |\mathbf{y}|)$ to $\frac{\mathcal{O}(|\mathbf{x}| \cdot |\mathbf{y}|)}{d}$. Besides, to further reduce video memory consumption, we split hidden states into small pieces before sent into joiner, and recombine it for back-propagation during training.

As the latency loss is defined as marginal expectation over all expanded paths $\hat{y}$, *mergeable* is also a requirement to the latency loss definition, which means latency loss through path $\hat{y}$ may be defined as $l(\hat{y}) = \sum_{k=1}^{|\mathbf{x}|+|\mathbf{y}|} l(\hat{y}_k)$ and $l(\hat{y}_k)$ is independent of $\hat{y}_{j' \neq j}$. However, both Average Lagging (Ma et al., 2019) and Differentiable Average Lagging (Arivazhagan et al., 2019) do not meet this requirement. We hence introduce a novel latency function based on wait-0 as oracle latency as follows:

$$
\begin{aligned}
d(i, j) &= \frac{1}{|\mathbf{y}|} \max\left(i - \frac{j \cdot |\mathbf{x}|}{|\mathbf{y}|}, 0\right) \\
l(\hat{y}_k) &= \begin{cases} 0 & \text{if } \hat{y}_k = \phi \\ d(i_k, j_k) & \text{else} \end{cases}
\end{aligned} \tag{2}
$$

Where $i_k = sum_{k'=1}^{k} I(\hat{y}_{k'} = \phi)$ and $j_k = sum_{k'=1}^{k} I(\hat{y}_{k'} \neq \phi)$ denote READ and WRITE actions number before $\hat{y}_k$. The latency for the whole expanded path $\hat{y}$ can be defined as

$$l(\hat{y}) = \sum_{k=1}^{|\hat{\mathbf{y}}|} l(\hat{y}_k) \tag{3}$$

Based on Eq. (3) the expectation of latency loss through all expanded paths may be defined as :

$$\begin{aligned} \mathcal{L}_{latency}(x, y) &= \mathbb{E}_{\hat{y} \in H(x,y)} l(\hat{y}) \\ &= \sum_{\hat{y}} \Pr(\hat{y}|y, x) l(\hat{y}) \end{aligned} \tag{4}$$

Latency loss and its gradients can be calculated by the forward-backward algorithm, similar to Sequence Criterion Training in ASR (Povey, 2005).

At last, we add the cross entropy loss of offline translation model as an auxiliary loss to CAAT model training for two reasons. First we hope the CAAT model fall back to offline translation in the worst case; second, CAAT models is carried out in accordance with offline translation when source sentence ended. The final loss function for CAAT training is defined as follows:

$$\begin{aligned} \mathcal{L}(x, y) &= \mathcal{L}_{CAAT}(x, y) + \lambda_{latency} \mathcal{L}_{latency}(x, y) \\ &+ \lambda_{CE} \mathcal{L}_{CE}(x, y) \\ &= -\log \sum_{\hat{y}} p(\hat{y}|x) \\ &+ \lambda_{latency} \sum_{\hat{y}} \Pr(\hat{y}|y, x) d(\hat{y}) \\ &- \lambda_{CE} \sum_{j} \log p(y_j|x, y_{<j}) \end{aligned} \tag{5}$$

Where $\lambda_{latency}$ and λ_{CE} are scaling factors corresponding to the $\mathcal{L}_{latency}$ and $\mathcal{L}_{CE}$. And we set $\lambda_1 = \lambda_2 = 1.0$ if not specified.

3.2 Streaming Encoder

Unidirectional Transformer encoder (Arivazhagan et al., 2019; Ma et al., 2020) is not effective for speech data processing, because of the closely related to right context for speech frame x_i. Block processing (Dong et al., 2019; Wu et al., 2020) is introduced for online ASR, but they lacks directly observing to infinite left context.

We process streaming encoder for speech data by block processing with right context and infinite left context. First, input representations $\mathbf{h}$ is divided into overlapped blocks with block step m and block size $m + r$. Each block consists of two parts, the main context $\mathbf{m}_n = [h_{m*n+1}, \cdots, h_{(m+1)*n}]$ and the right context $\mathbf{r}_n = [h_{(m+1)*n}, \cdots, h_{(m+1)*n+r}]$. The query, key and value of block $\mathbf{b}_n$ in self-attention can be described as follows:

$$\mathbf{Q} = \mathbf{W}_q [\mathbf{m}_n, \mathbf{r}_n] \tag{6}$$

$$\mathbf{K} = \mathbf{W}_k [\mathbf{m}_1, \cdots, \mathbf{m}_n, \mathbf{r}_n] \tag{7}$$

$$\mathbf{V} = \mathbf{W}_v [\mathbf{m}_1, \cdots, \mathbf{m}_n, \mathbf{r}_n] \tag{8}$$

By reorganizing input sequence and designed self-attention mask, training is effective by reusing conventional transformer encoder layers. And unidirectional transformer can be regarded as a special case of our method with $\{m = 1, r = 0\}$. Note that the look-ahead window size in our method is fixed, which ensures increasing transformer layers won't affect latency.

3.3 Text-to-Text Simultaneous Translation

We implemented both CAAT in Sec. 3.1 and wait-k (Ma et al., 2019) systems for text-to-text simultaneous translation. Hyper-parameters of our CAAT model architectures are shown in Table 6. CAAT training requires significantly more GPU memory than conventional Transformer (Vaswani et al., 2017), for the $\mathcal{O}\left(\frac{|x|\cdot|y|}{d}\right)$ complexity of joiner module. We mitigate this problem by decrease joiner hidden dimmension for lower decision step szie d.

3.4 Speech-to-Text Simultaneous Translation

3.4.1 End-to-End Speech-to-Text simultaneous Translation

The main system of End-to-End Speech-to-Text simultaneous Translation is based on the aforementioned CAAT structure. For speech encoder, two 2D convolution blocks are introduced before the stacked 24 Transformer encoder layers. Each convolution block consists of a 3-by-3 convolution layer with 64 channels and stride size as 2, and a ReLU activation function. Input speech features are downsampled 4 times by convolution blocks and flattened to 1D sequence as input to transformer layers. Other hyper-parameters are shown in Table 6. The latency-quality trade-off may be adjusted by varying the decision step size d and the latency scaling factor $\lambda_{latency}$. We submitted systems with best performance in each latency region.

	Parameters	S2T config	T2T config-A	T2T config-B
Encoder	layers	24	12	12
	attention heads	8	16	16
	dim_{ffn}	2048	4096	4096
	dim_{model}	512	1024	1024
	main context	32	/	/
	right context	16	/	/
Predictor	layers	6	6	6
	attention heads	8	16	16
	dim_{ffn}	2048	4096	4096
	dim_{model}	512	1024	1024
	dim_{output}	512	512	1024
Joiner	layers	6	6	6
	attention heads	8	8	16
	dim_{ffn}	1024	2048	4096
	dim_{model}	512	512	1024
/	dropout	0.1	0.1	0.1
	decision step size	{16,64}	{4,10,16,32}	{10,32}
	latency scaling factor	{1.0,0.2}	{1.0,0.2}	0.2

Table 6: Hyper-parameters configuration of End-to-End S2T and T2T simultaneous translation

3.4.2 Cascaded Systems

The cascaded system consists of two modules, simultaneous automatic speech recognition (ASR) and simultaneous text-to-text translation (MT). Both simultaneous ASR and MT system are built with CAAT proposed in Sec. 3.1. And we found the cascaded systems outperforms end-to-end system in medium and high latency region, though we cannot build cacaded system in low latency.

3.5 Unsegmented Data Processing

To deal with unsegmented data, we segment the input text based on sentence ending marks for T2T track, and for S2T task input speech is segmented to utterances with duration 20 seconds. Streaming of segmented pieces are directly sent to conventional simultaneous translation systems.

4 Experiment Results

4.1 Effectiveness of CAAT

To demonstrate the effectiveness of CAAT architecture, we compare it to wait-k with speculative beam search (SBS) (Ma et al., 2019; Zheng et al., 2019), one of the previous state-of-the-art. The latency-quality trade-off curves on S2T and T2T tasks are shown in Figures 3(a) and 3(b). We can find that CAAT significantly outperforms wait-k with SBS, especially in low latency section($AL < 1000ms$ for S2T track and $AL < 3$ for T2T track).

4.2 Effectiveness of data augmentation

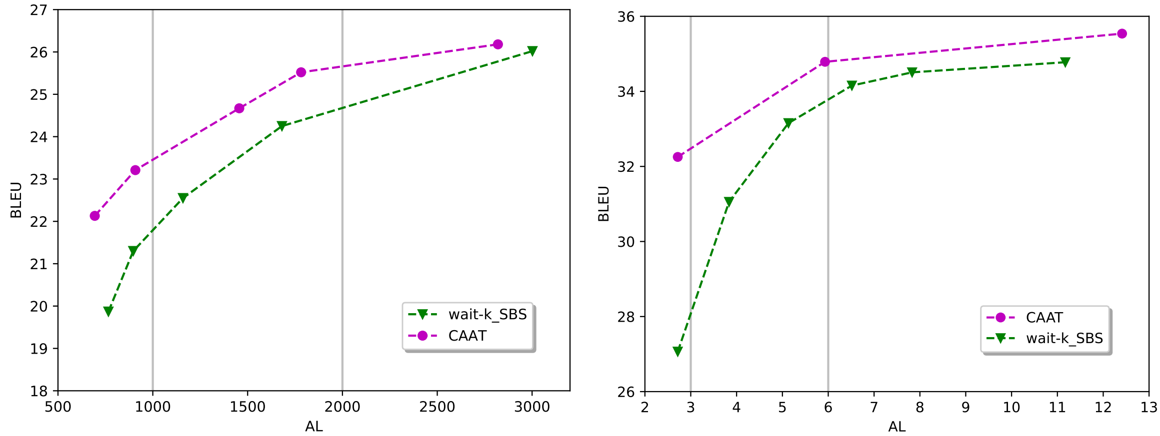
As illustrated in Table 7, adding new generated target sentences into the training corpora by using Self-training gives a boost of nearly 7 BLEU points and speech synthesis provides the other 1.5 BLEU points increase on the offline speech translation task, which usually determines the performance upper-bound of corresponding Speech-to-Text Simultaneous Speech Translation task.

Dataset	BLEU
Original speech corpora	21.24
+Seq-distill	28.21
+Seq-distill+Speech systhesis	29.72

Table 7: performance of offline speech translation on MuST-C_v2 tst-COMMON with different datasets

4.3 Text-to-Text Simultaneous Translation

EN→DE task. The performances of text-to-text EN→DE task is shown in the left sub-figure in Figure 4. As expected, the performance of proposed CAAT is always much better than that of wait-k with SBS and the best results from ON-TRAC (Elbayad et al., 2020) in 2020, especially in low latency regime, and the performance of CAAT with model-ensembling is nearly equivalent to offline result. Moreover, it can be further noticed from Figure 4(a) that the model-ensemble can also improve



(a) Speech-to-Text

(b) Text-to-Text

Figure 3: Translation quality against AL on EN→DE S2T and T2T simultaneous translation

the BLUE score more or less under different latency regimes, and the increase is quite obvious in low latency regime. Compared with the best result in 2020, we finally get improvement by 6.8 and 3.4 BLEU in low and high latency regime respectively.

En→JA task. Results of Text-to-Text simultaneous translation (EN→JA) track are plotted in the right sub-figure in Figure 4, where the curve naming CAAT_bst is best performances in this track with or without model-ensembling method. Curves in this sub-figure show the similar conclusion to the former subsection, that the result of proposed CAAT significantly outperforms that of wait-k with SBS. While we can also find that the gap between CAAT and offline is more obvious (nearly 0.4 BLEU), this is mainly because parameters of joiner block for EN→JA track in high-latency regime is reduced a lot from that for EN→DE track, due to the unstable EN→JA training.

4.4 Speech-to-Text Simultaneous Translation

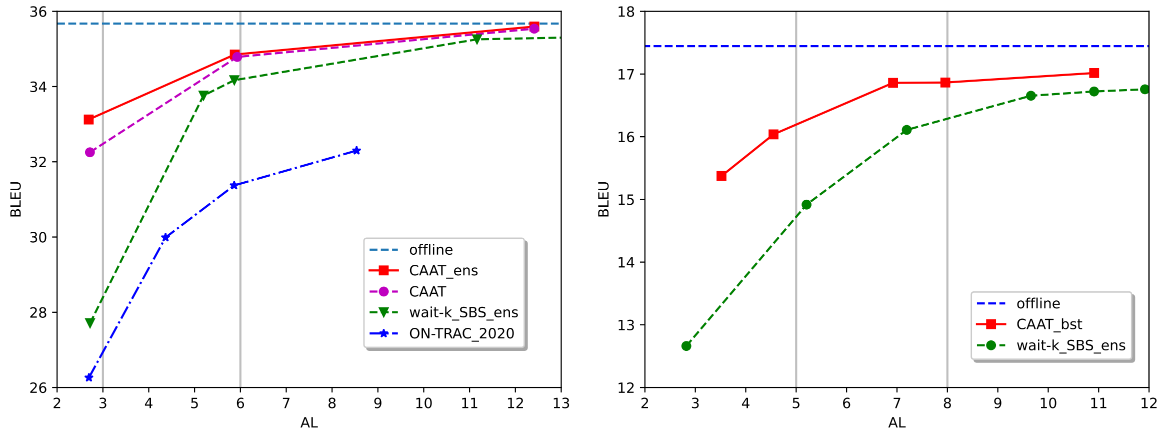
End-to-End System In this section, we discuss about our final results of End-to-End system based on CAAT. We tune the decision step size d and latency scaling factor $\lambda_{latency}$ to meet different latency regime requirements. For low, medium and high latency, the corresponding d and $\lambda_{latency}$ are set to (16,64,64) and (1.0,1.0,0.2) respectively. And during the training period, we sample the speech data from the whole corpora with the ratio MuST-C: TED-LIUM3: LibriSpeech: Europarl: CoVoST2: Speech synthesis dataset=2:2:1:1:1:5. We show our final latency-quality trade-offs in Figure 5. Combined with our data augmentation methods and new CAAT model structure, it can be seen that our single model system has already outperformed the best results of last year in all latency regimes and provides nearly 10 BLEU scores increase on average. Ensembling different models can further boost the BLEU scores by roughly 0.5-1.5 points.

Cascaded System Under the cascaded setting, we paired two well-trained ASR and MT systems, where the WER of ASR system's performance is 6.30 with 1720.20 AL, and the MT system is followed by the second configuration point indicated by CAAT_ens in Figure 4(a), whose results are 34.79 BLEU and 5.93 AL. We found that the best medium and high-latency systems are tuned with step size pair (d_{asr}, d_{mt}) as (6, 10) and (12, 10) respectively. And it turns out that the last cascading strategy described in Sec. 3.4.2 with well-assigned step size range $(lower, upper)$ provides the best performance, and here the number are (2, 8) and (4, 12) for medium and high-latency systems respectively.

Due to inherent latency bottleneck, we only provide results of medium and high latency systems, while the best of these two systems outperform our end-to-end counterparts by 0.29 and 0.78 BLEU respectively, with even lower AL. All the results in this section are evaluated on MuST-C_v2 tst-COMMON, and some of them are shown in Figure 5.

5 Conclusion

In this paper, we propose a novel simultaneous translation architecture, Cross-Attention Aug-



(a) Result of MuST-C_v2 tst-COMMON on EN→DE track (b) Result of IWSLT dev2021 on EN→JA track

Figure 4: Latency-quality trade-offs of Text-to-Text simultaneous translation

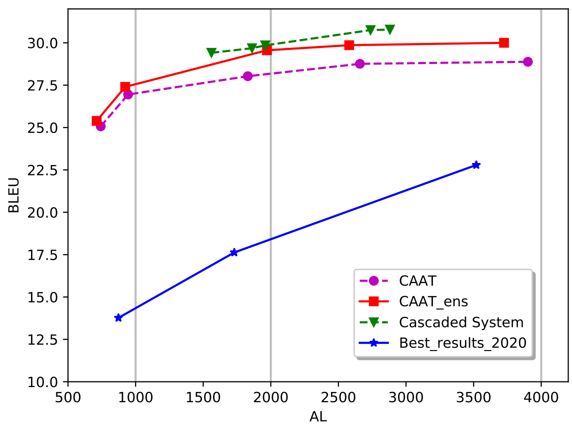


Figure 5: Latency-quality trade-offs of Speech-to-text simultaneous speech translation evaluated on MuST-C_v2 tst-COMMON

mented Transducer (CAAT), which significantly outperforms wait-k in both S2T and T2T simultaneous translation task. Based on CAAT model and data augmentation, we built simultaneous translation systems on three tasks, EN→DE S2T, EN→DE T2T task and EN→JA T2T task. We also built a cascaded speech-to-text simultaneous translation system for comparison. Both T2T and S2T systems are significant improvements over last year's best-performing systems.

References

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 Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Linhao Dong, Feng Wang, and Bo Xu. 2019. Self-attention aligner: A latency-control end-to-end model for ASR using self-attention network and chunk-hopping. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 5656–5660. IEEE.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Maha Elbayad, Ha Nguyen, Fethi Bougares, Natalia Tomashenko, Antoine Caubrière, Benjamin Lecouteux, Yannick Estève, and Laurent Besacier. 2020. On-trac consortium for end-to-end and simultaneous speech translation challenge tasks at iwslt 2020. *arXiv preprint arXiv:2005.11861*.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*.

Yoon Kim and Alexander M Rush. 2016a. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Yoon Kim and Alexander M. Rush. 2016b. Sequence-level knowledge distillation.

Taku Kudo. 2006. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab.sourceforge.jp*.

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: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yuchen Liu, Hao Xiong, Zhongjun He, Jiajun Zhang, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-end speech translation with knowledge distillation.

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 Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020. Monotonic multihead attention. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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. *arXiv preprint arXiv:1904.08779*.

Daniel Povey. 2005. *Discriminative training for large vocabulary speech recognition*. Ph.D. thesis, University of Cambridge.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, CONF. IEEE Signal Processing Society.

Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. SimulSpeech: End-to-end simultaneous speech to text translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3787–3796, Online. Association for Computational Linguistics.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE.

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.

Chunyang Wu, Yongqiang Wang, Yangyang Shi, Ching-Feng Yeh, and Frank Zhang. 2020. Streaming transformer-based acoustic models using self-attention with augmented memory. *arXiv preprint arXiv:2005.08042*.

Renjie Zheng, Mingbo Ma, Baigong Zheng, and Liang Huang. 2019. Speculative beam search for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1395–1402, Hong Kong, China. Association for Computational Linguistics.