Findings of the Second Workshop on Automatic Simultaneous Translation

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, Haifeng Wang
Baidu Inc. No. 10, Shangdi 10th Street, Beijing, 100085, China
{zhangruiqing01, zhangchuanqiang, hezhongjun, wu_hua}@baidu.com

Abstract

This paper presents the results of the shared task of the 2nd Workshop on Automatic Simultaneous Translation (AutoSimTrans). The task includes two tracks, one for text-to-text translation and one for speech-to-text, requiring participants to build systems to translate from either the source text or speech into the target text. Different from traditional machine translation, the AutoSimTrans shared task evaluates not only translation quality but also latency. We propose a metric "Monotonic Optimal Sequence" (MOS) considering both quality and latency to rank the submissions. We also discuss some important open issues in simultaneous translation.

1 Introduction

Simultaneous translation is to translate concurrently with the speech in the source language, aiming to obtain high translation quality with low latency. The concurrent comprehension and production process makes simultaneous translation an extremely challenging task for both human experts and machines. As a combination of machine translation (MT), automatic speech recognition (ASR), and text-to-speech synthesis (TTS), simultaneous translation still facing many problems to be studied in the research and application. To promote the development in this cutting-edge field, we conduct a shared task at the 2nd Workshop on Automatic Simultaneous Translation.

This year, we focus on Chinese-English simultaneous translation and set up two tracks:

1. **Text-to-text track**, where the participants are asked to submit systems that translate streaming input text in real-time. The input of this track is human-annotated transcripts in streaming format, in which every n-word sentence is broken into n lines of sequences whose length ranges from 1 to n, incremented by 1. We set up this track for two reasons. On the one hand, the difficulty of the task is reduced by removing the recognition of speech. On the other hand, participants can focus on text processing, such as segmentation and translation, without being influenced by ASR errors.

2. **Speech-to-text track**, where the submitted systems need to produce a real-time translation of the given audio.

We provide BSTC (Zhang et al., 2021) (Baidu Speech Translation Corpus) as the training data, which consists of about 68 hours of Mandarin speeches, together with corresponding transcripts, ASR results, and translations. In addition, participants can also use bilingual corpus provided by CCMT (China Conference on Machine Translation)[1]. We will describe the data in detail in Section 2.

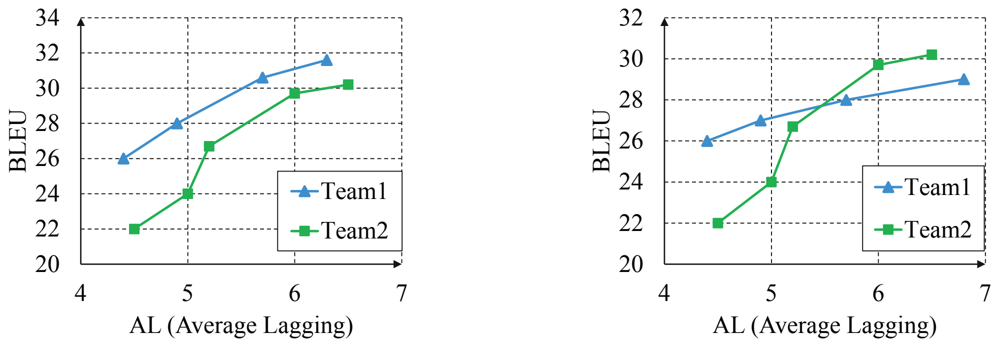
One objective of the shared task is to explore the performance of state-of-the-art simultaneous translation systems. Traditional evaluation metrics, such as BLEU, only measure the translation quality, while recently proposed metrics, such as Consecutive Wait (CW) (Gu et al., 2017) and Average Lagging (AL) (Ma et al., 2019) focus on latency. So far as we know, there is no metric that evaluates both quality and delay.

We ask the participants to submit systems under different configurations to produce multiple translation results with varying latency. Then we plot each result in a quality-latency coordinate. Normally, a system is regarded as the best if all of its points are above others (Figure 1(a)). However, in most cases, their lines of points intersect with each other (Figure 1(b)).

To consider both quality and latency in ranking, we propose a ranking metric, Monotonic Optimal Sequence (MOS) (Section 3). The idea is to first

[1] http://sc.cipsc.org.cn/mt/conference/2021/

Proceedings of the 2nd Workshop on Automatic Simultaneous Translation, pages 36–44
June 10, 2021. ©2021 Association for Computational Linguistics



(a) (b)

Figure 1: Two examples of the results submitted by two teams. Each point shows the latency (X-axis) - BLEU (Y-axis) of a submitted system.

Corpus		Train	Dev	Test
BSTC	Audio (hours)	64.6	1.6	1.5
	#Talks	215	16	6
	#Utterances	37,901	956	975
CCMT	#Sentence Pairs	9.1M	2,000	/

Table 1: The summary of our provided corpora. The Dev set of CCMT2020 is Newstest2019. There are multiple test sets for CCMT so we don't list the statistics.

find all the optimal points, that is, a group of points with the highest quality under different latency, and then calculate the proportion of a system's optimal points in all its submitted points. The higher the proportion, the better the performance.

We received six submissions from four teams this year. We will report the results and analysis in Section 4. We discuss some important open issues in Section 5 and conclude the paper in Section 6.

2 Shared Task

We first introduce the data sets used in the shared task and the setup of the two tracks.

2.1 Training Set

Due to the scarcity of Zh→En speech translation corpora, we provide a Zh→En speech translation dataset BSTC and a large-scale text translation corpus CCMT for the participants.

- **BSTC** (Zhang et al., 2021) (Baidu Speech Translation Corpus) is a 68-hour Zh→En speech translation data including 215 speeches for training. Each speech is segmented into sentences, transcribed, and translated into English.
- We also encourage participants to use the large-scale Zh→En text translation corpus **CCMT 2020** (China Conference on Machine Translation) to enhance the performance of machine translation.

The statistics of the two datasets are listed in Table 1. As far as we know, BSTC is by far the largest Zh→En speech translation corpus, but it is still insufficient to train either a well-performed ASR model or an end-to-end simultaneous translation model in the speech-to-text track. Therefore, we don't impose restrictions on the dataset used by the participants for the speech track.

2.2 Test Set

Notice that the test set of BSTC shown in Table 1 is not released. The participants are required to submit docker systems, which will be tested on the 1.5-hours test set by us.

The test set is kept confidential as a progress test set. To validate the system to submit, we provide the dev set to the participants, which has the same format as the test set. It contains four-way parallel samples of 1) the streaming transcript, 2) the streaming asr, 3) the sentence-level translation of the transcript, and 4) the audio. The streaming transcripts are produced by turning each n-word (a word means a Chinese character here) sentence to n lines of word sequences with length 1, 2, ..., n. And the streaming ASR is produced by the real-time Baidu ASR system based on SMLTA[2].

2.3 Two Tracks

We set two tracks in our shared task, the text-to-text track is to input streaming transcripts and the

[2] http://research.baidu.com/Blog/index-view?id=109

speech-to-text track is to input audio files, as mentioned in section 1.

The simultaneous translation aims to balance system delay and translation quality. The key problem is to explore a policy that decides when to begin translating a source sentence before the speaker has finished his/her utterance. Eager policies, such as translating every word when it is received, will lead to poor translation quality, while lazy policies, such as waiting to translate until receiving a complete sentence, will result in long system delay.

In order to comprehensively evaluate each system's performance, we suggest that the participants generate multiple results on varying latency. Six systems from four teams were submitted in the shared task, four to Track 1 and two to Track 2.

3 System Evaluation

Unlike text translation evaluation that only takes one indicator (i.e., translation quality), simultaneous translation evaluation needs to consider quality and latency at the same time. The evaluation based on two criteria brings difficulties to ranking the systems. However, the two indicators are not easy to merge into one.

To rank the submissions better, we propose a ranking algorithm called Iterative Monotonic Optimal Sequence (I-MOS). Specifically, we define an *optimal point* as the result of the best translation quality at each latency. Our algorithm iteratively finds sets of optimal points to construct an optimal curve called Monotonic Optimal Sequence (MOS), then each team's proportion of points on the MOS curve is calculated to measure the performance. The overall process is illustrated in Figure 2.

In the following sections, we first introduce the commonly used metrics of quality and latency (Section 3.1), then propose the Monotonic Optimal Sequence (Section 3.2) and elaborate our I-MOS algorithm (Section 3.3).

3.1 Evaluation metrics

In simultaneous translation, quality is often measured by BLEU (Papineni et al., 2002). Recent work proposed some metrics for latency evaluation, such as Average Proportion (AP) (Cho and Esipova, 2016), Consecutive Wait (CW) (Gu et al., 2017), Average Lagging (AL) (Ma et al., 2019) and Differentiable Average Lagging (DAL) (Arivazhagan et al., 2019). Here we briefly introduce the two latency metrics used in our evaluation:

- **CW** is the average source segment length in words. It measures the number of source words being waited for between each two translation actions.

- **AL** quantifies the degree the audience is out of sync with the speaker by the average number of source words that the audience lags behind the ideal policy, in which the translation of each sentence is output at the same speed as the speaker's utterance and the entire translation finished when the speaker completes his/her utterance.

Note that the above-mentioned latency metrics are all proposed for text-to-text simultaneous translation and we use AL in the text track for latency evaluation. Some work extended AP and AL to speech translation (Ren et al., 2020; Ma et al., 2020), but we don't use them because they measure real-time latency, while some submissions calling remote services contain network delay. It is unreasonable to use real-time latency metrics for both the local-running systems and remote-running systems. Thus we ignore the latency of the ASR model and take the metrics of text-to-text simultaneous translation in the speech track. Specifically, we use BLEU-AL evaluation in the Text-to-text track and BLEU-CW evaluation in the Speech-to-text track.

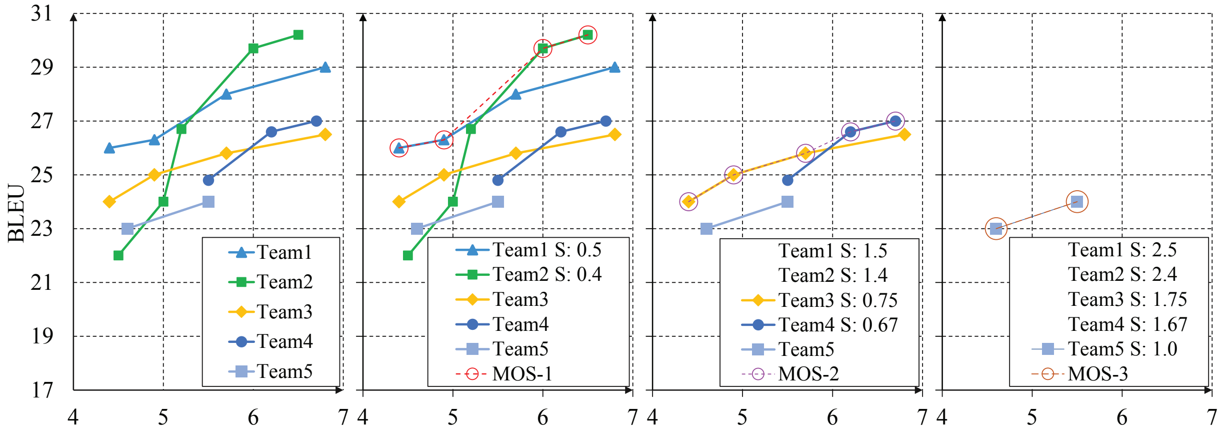
3.2 Monotonic Optimal Sequence

To comprehensively rank systems based on the translation quality and latency, we propose to construct a monotonic optimal sequence composed of *Optimal Point*s.

Definition 1. On the quality-latency figure, one result is considered optimal if there is no other point or line above it at an identical latency. In this case, the result is of the highest translation quality at that latency and we define it as an ***Optimal Point***.

For example, among the nine results of Figure 1 (b), the leftmost two points of Team1 and rightmost two points of Team2 are *Optimal Point*s. The third point from left on Team2's curve is not optimal because it lies below the line of Team1.

To get *Optimal Point*s, we select the results of the best translation quality with different latency. Since the submitted systems have discrete latency, we use the linear interpolation of adjacent points of each team to estimate their translation quality on continuous latency. Then we select some *Optimal Point*s to form an optimal curve called Monotonic Optimal Sequence.



(a). Plotting all teams (b). Step 1 of I-MOS (c). Step 2 of I-MOS (d). Step 3 of I-MOS

Figure 2: An illustration of our Iterative Monotonic Optimal Sequence (I-MOS) algorithm. First (a). plot the results of all teams, then (b) (c) (d) iteratively calculate the monotonic optimal sequence (MOS) of level k and update the score of the teams belong to level 1, 2, ..., k. The X-axis denotes the average lagging.

Definition 2. Let **Monotonic Optimal Sequence** (MOS) be a sequence of *Optimal Point* with increasing translation quality and latency.

We arrange all the *Optimal Points* in ascending order of latency and then select the points with monotonously increasing translation quality to form the MOS. The monotonicity requirement for translation quality is to avoid outlier points. For example, the rightmost point of Team1 in Figure 2 (b) is an outlier because there is no point or line above this point at the same latency, but it doesn't follow the monotonicity principle, so it should not be added to MOS.

We propose to use each team's proportion of points on the MOS to evaluate its performance. That is, we rank teams with:

$$S_{T_i} = \mathcal{N}(p^*_{t_i})/\mathcal{N}(p_{t_i}) \qquad (1)$$

where $\mathcal{N}(p^*_{t_i})$ and $\mathcal{N}(p_{t_i})$ denote the number of points on MOS and the number of submitted points of team i, respectively. Therefore, the maximum value of S_{T_i} is 1, when all of its submitted points are on the MOS.

3.3 Iterative Monotonic Optimal Sequence Algorithm

There exists a problem in our measurement that, according to Eq. 1, all the teams that have no points on the MOS are ranked tied because they all score zero. To tackle this problem, we propose the Iterative Monotonic Optimal Sequence (I-MOS) algorithm. The main idea is to iteratively calculate the MOS curves, MOS-1, MOS-2, ... MOS-K, in which MOS-k denotes the Monotonic Optimal Sequence of level k calculated at the k^{th} iteration. All the systems that have at least one point on MOS-k are classified to level k. We remove these systems and calculate MOS-$(k+1)$ in the next iteration. Each team of the k^{th} level ranks higher than all teams of the $(k+1)^{th}$ level.

Our algorithm is elaborated in Algorithm 1. The level of all teams is initialized to zero (line 1), which denotes the team's score has not been calculated. Then we begin our iteration. While there exists at least one team whose score has not been calculated (line 4), we update the score of teams that belong to superior levels (level $1, 2, ..., k-1$) teams by adding the maximum value of S_{T_i} (1 point) to them (line 5-7) to ensure the systems of level $1, 2, ...k-1$ scores higher than systems of level k. Then we calculate MOS-k (line 8) and update the score of the teams that belong to level k according to Eq. 1 (line 9-11). After an iteration, we continue to explore teams that belong to level $k+1$ (line 12). Figure 2 provides a running process of I-MOS.

4 Systems Results

We received 6 systems submitted by four teams from four universities:

- Institute of computing technology, Chinese Academy of Science (ICT)
- Xiamen University (XMU)
- Beijing Institute of Technology (BIT)
- Ping An Technology (Shenzhen) Co., Ltd. (PingAn)

Algorithm 1: Iterative Monotonic Optimal Sequence (I-MOS)

```
Input: Number of teams $N$
Input: Teams submission: $t_i$ contains all
       results submitted by team $i$
Output: Teams score $S$: $s_i$ is the score of
        team $i$ for ranking
$tl = [0, 0, ..., 0]$      ▷ Initialize teams level
      ▷ $tl[i]$ denotes the level of team $i$
$k \leftarrow 1$           ▷ Start from level 1
while $\prod_{i=1}^{N} tl[i]$ = 0 do
    for $i$=1, 2, ..., $N$ do
        if $tl[i] \neq 0$ then
            $s[i] \leftarrow s[i] + 1$
    Calculate MOS-$k$   ▷ the $k^{th}$ level MOS
    for $i$=1, 2, ..., $N$ do
        if $t_i$ has at least one point on MOS-k then
            $tl[i] \leftarrow k$
            $s[i] \leftarrow \mathcal{N}(p^*_{t_i})/\mathcal{N}(p_{t_i})$
    $k \leftarrow k + 1$
```

Track 1		
Team Level	**Team**	$\mathcal{N}(p^*_{t_i})/\mathcal{N}(p_{t_i})$
Level 1	ICT	4/4
	XMU	2/3
	BIT	1/4
Level 2	PingAn	7/7
Track 2		
Level 1	PingAn	1/1
	XMU	1/3

Table 2: The evaluated level of each team and the proportion of points on the MOS of the corresponding level. The table shows the ranking of the teams from top to bottom.

We test each docker system with our testset, which contains 1.5 hours of 6 Mandarin talks. All the systems are run on V100 GPU. We plot the evaluation results in Figure 3 and rank them according to the I-MOS algorithm. Their ranking results are shown in Table 2. We use BLEU[3] to evaluate the translation quality and use Average Lagging (AL) (Ma et al., 2019) and Consecutive Wait (CW) (Gu et al., 2017) as latency metrics.

4.1 Text-to-text Track

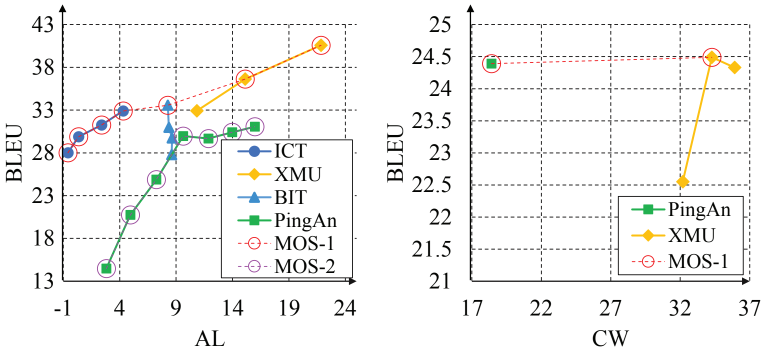
In the first track, the results of the four teams reflect their preference in balancing system latency and translation quality. We briefly describe the methods of the four teams below in the order of their ranks:

1. **ICT** proposes the character-level *wait-k* policy, rather than using the standard word-level *wait-k* (Ma et al., 2019). They perform prefix-to-prefix MT training as in the original work. Besides, they follow the *multi-path* (Elbayad et al., 2020) and *future-guided* (Zhang et al., 2020b) methods to enhance the predictability and avoid huge anticipation in translation caused by *wait-k*. The *multi-path* method adopts randomly sampled k in $[1, 2, ..., K]$ in the training of incremental MT model to cover all possible k during training. And the *future-guided* method attempts to promote the prediction ability of the *wait-k* strategy. To improve the robustness of the MT model, they further try several data augmentation methods via adding noise to the source text.

2. **XMU** follows the *Meaningful Unit* (MU) segmentation policy proposed in Zhang et al. (2020a) that uses a context-aware classification model to determine whether the currently received ASR content can be definitely translated. To generate consistent translation given the segmentation, the MT model of the pipeline system is used to automatically generate training data of meaningful units. The MT model is trained by full-sentences pairs.

3. **BIT** uses a pipeline method with a segmentation model that bridges the streaming text input and the MT model. Once a punctuation mark is detected, the segmentation sends the currently received sub-sentence for translation as in (Zhang and Zhang, 2020). To make the MT model adapt to translating short sub-sentences at inference time, each sample in the provided parallel training corpus is automatically divided into multiple translation pairs for training. A statistical word alignment tool is used to segment the source sentence into minimal chunks so that crossing alignment links between source and target words occur only within individual chunks. The parallel pairs of chunks are then used to train their MT model.

[3] BLEU is calculated using " https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v13a.pl".



(a). Track1. Text-to-Text AST

(b). Track2. Speech-to-Text AST

Figure 3: The evaluation results of the two tracks. The order in the legend denotes the real ranking.

4. **PingAn** takes the test-time *wait-k* (Ma et al., 2019) as the segmentation policy. Different from the standard *wait-k* policy, test-time *wait-k* uses the *wait-k* policy only at inference time without prefix-to-prefix training the MT model. They further adopt *Back-Translation* (Sennrich et al., 2016) to improve the translation quality.

In summary, we can categorize the four systems according to their segmentation policy: Both **ICT** and **PingAn** adopt the *wait-k* policy. **ICT** adopts training-time *wait-k* while **PingAn** uses test-time *wait-k*. **BIT** chooses sub-sentence translation, that is, to translate only when a punctuation is detected. **XMU** performs *MU-based* segmentation in which the training samples of meaningful units are generated by the MT model.

Figure 3 (a) shows that the latency of the two methods using *wait-k* is relatively low, while *MU-based* policy can achieve high translation quality. For the two *wait-k* systems, **ICT** performs better than **PingAn**, which is consistent with the experimental results in Ma et al. (2019) that training-time *wait-k* is superior to test-time *wait-k*.

It's interesting to find that the latency of **XMU** is larger than that of **BIT**. This might be because there are often long-distance reorderings in the training corpus. The reordering in translation that crosses punctuation marks would prevent the *MU* segmentation policy from extracting fine-grained MUs, resulting in the average length of *MU*s exceeding sub-sentences. This problem has been illustrated in Zhang et al. (2020a) and they proposed a refined method called *MU++* to alleviate the problem.

The result of **BIT** is a little weird. The translation quality decreases as system latency grows. This might be caused by the discrepancy between the segmentation module and the MT model. In their method, the segmentation module segments sentences into sub-sentences while the MT model is trained on statistically split chunks.

4.2 Speech-to-text Track

As elaborated in Section 3.1, we use BLEU and Consecutive Wait (CW) (Gu et al., 2017) to evaluate systems in the speech track.

PingAn and **XMU** continue their work based on their systems submitted to the Text-to-text track. The two systems both keep the same policy used in the first track and only replace the text input with the recognition results of an ASR model. **PingAn** trains a QuartzNet model (Kriman et al., 2020) with the Memory-Self-Attention (Luo et al., 2021) and **XMU** uses Baidu's real-time speech recognition service.

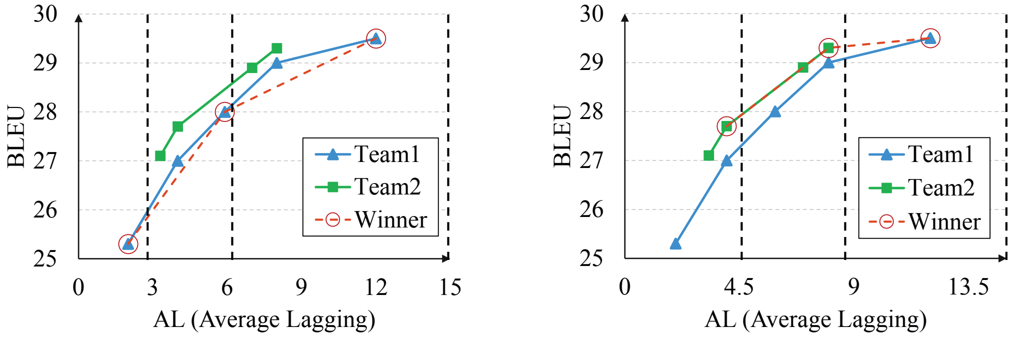
Figure 3 (b) shows that **PingAn** using wait-k outperforms **XMU** in latency. The reason behind the large delay of **XMU**'s system might be the same as in the first track.

5 Discussion

Most recent studies on simultaneous translation focused on methods to balance translation quality and latency. Besides this, we will discuss some other important challenges for simultaneous translation.

5.1 Data Scarcity

The first problem is the shortage of high-quality simultaneous translation data. In recent years, some speech translation corpora have released, such as MuST-C (Di Gangi et al., 2019), Covost (Wang et al., 2020a,b), Europarl-ST (Iranzo-Sánchez et al., 2020), Aug-LibriSpeech (Kocabiyikoglu et al., 2018), etc. These corpora focus on Indo-European



(a). IWSLT's Ranking with regimes boundary (3, 6, 15)

(b). IWSLT's Evaluation with regimes boundary (4.5, 9, 13.5)

Figure 4: An illustration of the ranking algorithm of IWSLT's simultaneous translation shared task. The two figures vary only in the threshold of the latency regimes. According to their algorithm, the winner of figure (a) is Team1 in all the three regimes, while the winner evaluated in figure (b) is *Low Latency*: Team2, *Medium Latency*: Team2, and *High Latency*: Team1.

languages and have greatly contributed to the increasing popularity of research of simultaneous translation.

However, there is little attention paid to research and data collection of Chinese-English (Zh→En) simultaneous translation. To the best of our knowledge, only MSLT (Federmann and Lewis, 2016) and Covost (Wang et al., 2020b) contain Zh→En speech translation data, but they totally have about 30 hours of speech. In our shared task, we build 68-hour Zh→En speech translation corpus, BSTC (Zhang et al., 2021) for training and evaluation. The dataset alleviates the Zh→En data scarcity, but it's still insufficient to train data-hungry end-to-end simultaneous translation models.

5.2 Evaluation Dilemma

The second problem lies in system evaluation, which has not been widely explored.

Traditional metrics such as BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), etc, are designed for text translation. These metrics based on accurate matching between system outputs and references. However, to reduce latency in simultaneous interpretation, human interpreters usually use strategies such as reasonable omissions, avoiding long-distance reordering in translation, etc. Thus the traditional metrics are not suitable to evaluate the simultaneous interpretation.

On the other hand, there is no metric to evaluate both translation quality and latency. In our shared task, we propose a novel ranking algorithm, I-MOS. We only consider the proportion of *optimal points*, ignoring whether the points lie in low-latency or high-latency. Therefore, our ranking doesn't differentiate latency regimes. However, it remains open to question whether it is reasonable to compare two systems with no intersection in latency, like the **ICT** and **XMU** in Figure 3 (a). The ranking might be more convincing if **ICT** had provided results at high latency and **XMU** has provided results at low latency.

We note that IWSLT has also hosted simultaneous translation shared tasks[4]. They proposed to rank systems by the translation quality with different latency regimes: *Low Latency*: AL <= 3, *Medium Latency*: AL <= 6, and *High Latency*: AL <= 15. For each team, the submitted system that achieves the best translation quality is chosen for ranking in each latency regime. However, the value of artificially defined latency threshold between regimes has a big impact on the ranking results. As illustrated in Figure 4, different latency thresholds lead to completely different rankings of the two teams.

Actually, the ideal ranking mechanism is to rank all systems within a similar latency interval. However, asking participants to submit results in almost every latency regime is unreasonable, because existing policies all have a preference in trading off latency and translation quality. For example, *wait-k* focuses on getting controllable low latency, while the inspiration behind *MU* is to translate until a segment with definite meaning is formed, leading to a high latency as well as high quality. Therefore, it is a dilemma to evaluate systems comprehensively while distinguishing different latency regions reasonably. This problem can be explored in future

[4] https://iwslt.org/2021/simultaneous

work.

5.3 Applications

Recently, more and more simultaneous translation systems have emerged in international conferences.

In practical applications, systems face robust and controllability issues. Being robust denotes the system should achieve a high translation quality and be insensitive to speech noise, including sound capture noise, speaker's accent, disfluency in speech, etc. Being controllable means the system should be able to remember and understand some named entities and should be able to be intervened.

Our shared task provides such an opportunity for participants to pay attention to the robustness problem. For example, **ICT** and **PingAn** have adopted data augmentation to enhance the robustness of their systems.

In terms of controllability, it is not difficult to integrate an intervention mechanism in pipeline systems. For example, a pre-defined translation of a named entity can be introduced to the MT module. However, controllability is not easy to be guaranteed for end-to-end simultaneous translation systems (Ren et al., 2020; Ma et al., 2020). It remains a challenge to correct a translation without an intermediate ASR result. We also hope to see more work focusing on real-world simultaneous translation applications and discussing some interesting issues, such as the document-level ASR error correction in pipeline systems, and how to enhance the controllability in end-to-end speech-to-text systems, etc.

6 Conclusion

This paper presents the results of the Zh→En simultaneous translation shared task hosted on the 2nd Workshop on Automatic Simultaneous Translation (AutoSimTrans). The shared task includes two tracks, the text-to-text track (Track1) and the speech-to-text track (Track2). Six systems were submitted to the shared task, four to Track1 and two to Track2. We propose an evaluation method "Monotonic Optimal Sequence" (MOS) to evaluate both translation quality and time latency. We report the results and further discuss some important open issues of simultaneous translation.

Regrettably, the number of submissions is less than expected, especially for the speech-to-text track. In fact, there are more than 300 teams registered. However, most of them did not submit their results. The possible reason may be that the interdisciplinary task is not easy for participants. We hope to see more participants in the future.

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.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *arXiv preprint arXiv:1606.02012*.

Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. Must-c: a multilingual speech translation corpus. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2012–2017. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient wait-k models for simultaneous machine translation. *arXiv preprint arXiv:2005.08595*.

Christian Federmann and William D Lewis. 2016. Microsoft speech language translation (mslt) corpus: The iwslt 2016 release for english, french and german. In *International Workshop on Spoken Language Translation*.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor OK Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233. IEEE.

Ali Can Kocabiyikoglu, Laurent Besacier, and Olivier Kraif. 2018. Augmenting librispeech with french translations: A multimodal corpus for direct speech translation evaluation. *Language Resources and Evaluation*.

Samuel Kriman, Stanislav Beliaev, Boris Ginsburg, Jocelyn Huang, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, and Yang Zhang. 2020. Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6124–6128. IEEE.

Jian Luo, Jianzong Wang, Ning Cheng, and Jing Xiao. 2021. Unidirectional memory-self-attention transducer for online speech recognition. *arXiv preprint arXiv:2102.11594*.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, et al. 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.

Xutai Ma, Juan Pino, and Philipp Koehn. 2020. Simulmt to simulst: Adapting simultaneous text translation to end-to-end simultaneous speech translation. *arXiv preprint arXiv:2011.02048*.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, QIN Tao, 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu. 2020a. Covost: A diverse multilingual speech-to-text translation corpus. *arXiv preprint arXiv:2002.01320*.

Changhan Wang, Anne Wu, and Juan Pino. 2020b. Covost 2: A massively multilingual speech-to-text translation corpus. *arXiv preprint arXiv:2007.10310*.

Ruiqing Zhang, Xiyang Wang, Chuanqiang Zhang, Zhongjun He, Hua Wu, Zhi Li, ying Chen, and Qinfei Li. 2021. Bstc: A large-scale chinese-english speech translation dataset. In *Proceedings of the Second Workshop on Automatic Simultaneous Translation*. Association for Computational Linguistics.

Ruiqing Zhang and Chuanqiang Zhang. 2020. Dynamic sentence boundary detection for simultaneous translation. In *Proceedings of the First Workshop on Automatic Simultaneous Translation*, pages 1–9, Seattle, Washington. Association for Computational Linguistics.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2020a. Learning adaptive segmentation policy for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2280–2289, Online. Association for Computational Linguistics.

Shaolei Zhang, Yang Feng, and Liangyou Li. 2020b. Future-guided incremental transformer for simultaneous translation. *arXiv preprint arXiv:2012.12465*.