XMU's Simultaneous Translation System at NAACL 2021

Shuangtao Li[1] and **Jinming Hu**[1]
and **Boli Wang**[2] and **Xiaodong Shi**[3]* and **Yidong Chen**[3]
School of Informatics, Xiamen University / Xiamen, China
[1]{lishuangtao,todtom@stu.xmu.edu.com}
[2]me@bo-li.wang
[3]{mandel,ydchen}@xmu.edu.cn

Abstract

This paper describes XMU's two systems submitted to the simultaneous translation evaluation at the 2nd automatic simultaneous translation workshop, which are for Zh->En text-to-text translation and Zh->En speech-to-text translation. In both systems, our translation model is based on Transformer. To translate streaming text, we use an adaptive policy to split the text into appropriate segments and translate them monotonically. Our speech-to-text system is a pipeline system, in which the MT component is exactly the same as our text-to-text system.

1 Introduction

Simultaneous translation refers to translating the message from the speaker to the audience in real-time without interrupting the speaker. It is widely used in many scenarios such as international conferences and business negotiations. Simultaneous machine translation is a challenging task and has become an increasingly popular research field in recent years.

There have been some researches on simultaneous translation of speech input (Niehues et al., 2018; Ma et al., 2020b,c; Ren et al., 2020), and some researches focused on text translation (Arivazhagan et al., 2019; Zhang et al., 2020; Ma et al., 2020a).

In this paper, we describe our two systems submitted to the simultaneous translation evaluation at the 2nd automatic simultaneous translation workshop, which are for Zh->En text-to-text translation and Zh->En speech-to-text translation. We build our systems with the state-of-the-art method (Zhang et al., 2020), and verify the effectiveness of this method.

2 Text-to-text Track

In this section, we describe our system submitted to Zh->En text-to-text simultaneous translation track.

The main idea of this system is how human interpreters work. While listening to speakers, human interpreters constantly translate text segments that are appropriate to translate without waiting for more words, and meanwhile making the translation grammatically tolerable. Text segments considered appropriate to translate usually have clear and definite meaning, because the translation of such a segment is not likely to be changed by subsequent text. The authors of Zhang et al. (2020) refered to such segments Meaningful Units (MU) and gived MUs a precise difinition. See Table 1 for an illustration.

Our system works like a human interpreter described above and is composed of an MU classification model and a machine translation model. Once a segment is classified to be an MU by the MU classifier, the MT model uses forced decoding with previous translation as the prefix to translate the segment, as shown in Table 1.

2.1 Machine translation

Our machine translation model is implemented with FAIRSEQ[1] (Ott et al., 2019).

Data and preprocessing. The data we use are CWMT19[2] (9.1M parallel sentences pairs) and the simultaneous translation corpus (39K parallel sentences pairs) provided by the organizer of the workshop.

We do the following steps to preprocess the data.

- Filtering. The sentence pairs whose English sentence is longer than 120 words are filtered out.

* Corresponding author.
[1]https://github.com/pytorch/fairseq
[2]http://mteval.cipsc.org.cn:81/agreement/description

Proceedings of the 2nd Workshop on Automatic Simultaneous Translation, pages 19–23
June 10, 2021.

Source:	牛顿 Newton	||	发现 discover	了 tense particle	||	牛顿 Newton	运动 motion	定律 law
Simul. Interpretation:	Newton	||		discovered	||			Newton’ s laws of motion

Table 1: An illustration of how a human interpreter work. The source sentence is splited to three MUs (separated by "||"), and an interpreter translates the MUs in order and makes them form a grammatically correct sentence.

- There are a few punctuation marks, numbers and letters in the data which are in full width. They are converted to half width characters.
- There are a few Chinese characters in the data which are traditional characters. They are converted to simplified ones.
- Chinese segmentation. All Chinese sentences are segmented with Jieba Chinese Segmentation Tool[3].
- English tokenization. All English sentences are tokenized and truecased with Moses[4].
- Byte-pair-encoding (BPE) (Sennrich et al., 2016). Both Chinese and English data are encoded by BPE with Subword-NMT[5]. The number of merge operations for each language is set to 30K.

Modeling and training. Our translation model's architecture is base Transformer (Vaswani et al., 2017). We use Adam optimizer (Kingma and Ba, 2015) to optimize the loss. We use weight decay of $1e^{-4}$ and dropout with probability of 0.2 for regularization. Label smoothing with ϵ of 0.1 is applied to our model. During inference, we set beam size to 20.

Our model is first pretrained on CWMT and then finetuned on the the training set of the Baidu Speech Translation Corpus (Zhang et al., 2021). We set learning rate to $5e^{-4}$ in the pretraining stage and $3e^{-5}$ in the fine-tuning stage. The learning rate is linearly increased for the first 4000 training steps, and is decreased following an inverse squareroot schedule.

2.2 MU classifier

Modeling and training. The MU classifier is a binary classifier. Given a source word sequence $x = \{x_1, x_2, ..., x_n\}$, the MU classifier determines whether x ends with an MU, and if so the MT will translate x with forced decoding. The input of the classifier is x and m "future" words $x_f = \{x_{n+1}, x_{n+2}, ..., x_{n+m}\}$, where m is a hyperparameter. The outputs are the probabilities of two classes $p(c = 1)$ and $p(c = 0)$, which mean x ends with an MU or not. x will be classified into class 1 if $p(c = 1) > t$, where t is a threshold set based on experience. Obviously, we can control the latency of simultaneous translation by modifying m. Later in experiments, it will be shown that we can also control the latency by modifying t. In our system, m is always set to 2.

x	$x_f(m = 2)$	c
牛顿 ||	发现 了	1
牛顿 || 发现	了 牛顿	0
牛顿 || 发现 了 ||	牛顿 运动	1
牛顿 || 发现 了 || 牛顿	运动 定律	0

Table 2: MU samples for the MU classifier. "||" is a symbol to separate MUs. $c = 1$ means that x ends with an MU, otherwise not.

The MU classifier is based on a chinese BERT (Devlin et al., 2019)[6]. We use the base model and fine-tune it with a learning rate of $5e^{-4}$.

Generating MU samples. To build an MU classifier, we need to generate MU samples just like the samples in Table 2. For each sentence of length N, we generate $N - m$ examples for it, and every MU sample is a triple $< x, x_f, c >$. When we generate examples, c is set to 1 if x ends with an MU, else it is set to 0. In our system, we generate MU samples for every sentence pairs in CWMT and the simultaneous translation corpus. Our MU samples are a little different from the MU samples in Zhang et al. (2020). In their work, the future words of a sample can be less than m, but not in this paper. We do not need training samples whose future words are less than m, because during inference when the future words are less than m, the sentence is already a whole sentence and thus can be fed into MT.

We use the *basic method* proposed in Zhang et al. (2020) to generate MU samples.

[3]https://github.com/fxsjy/jieba
[4]https://github.com/moses-smt/mosesdecoder
[5]https://github.com/rsennrich/subword-nmt
[6]https://github.com/649453932/Bert-Chinese-Text-Classification-Pytorch

3 Speech-to-text Track

In this section, we describe our system submitted to Zh->En speech-to-text simultaneous translation track.

This system is a pipeline of three stages: (1) speech recognition, (2) punctuation restoration, and (3) streaming machine translation. The third stage is exactly the same as the system described in Section 2. In other words, the system described in Section 2 is a part of our speech-to-text system, and therefore it will not be repeated in this section. This section only describes the stage (1) and stage (2).

3.1 Speech recognition

Instead of building a speech recognition model, we use Baidu's real time speech recognition service[7]. In our system, We call the API of this service to recognize streaming speech. It is important to note that although the ASR does not output punctuation, it separates different sentences, that is, the ASR outputs are many segmented sentences instead of one sentence.

3.2 Punctuation restoration

The recognition results of Baidu's asr service do not have punctuation, but the input of our MT model needs punctuation. As a result we build a model to restore the punctuation for every recognition result. We use a BERT-based (Devlin et al., 2019) sequence labeling model (Chen and Shi, 2020) to do punctuation restoration. This model labels every Chinese character in a sentence, and for the model we only consider four classes: comma, period, question mark and no punctuation.

4 Experiments

In this section, we evaluate our two systems on the development set of the Baidu Speech Translation Corpus (Zhang et al., 2021). The two used metrics are case-sensitive detokenized BLEU (Papineni et al., 2002) and Consecutive Wait (CW) (Neubig et al., 2017), for translation quality and latency respectively. CW considers on how many source words are waited for consecutively between two target words, and thus larger CW means longer latency. We use SacreBLEU (Post, 2018)to compute BLEU scores.

[7]https://cloud.baidu.com/doc/SPEECH/s/2k5dllqxj

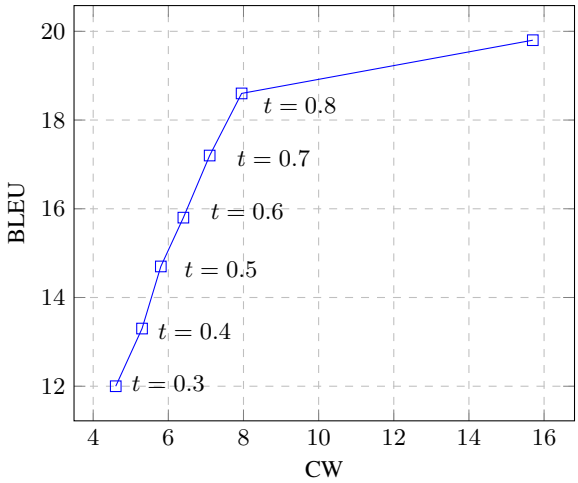

Figure 1: Translation quality against latency of different thresholds t. The rightmost point is not a result of simultaneous translation, but a result got by translating complete sentences.

4.1 Text-to-text track

We set the threshold t in the MU classifier to various values and get multiple results, as shown in Figure 1. It is worth noting that the rightmost point in Figure 1 is not a result of simultaneous translation. This result is got by translating every sentence after it is finished, i.e., we get this result by translating whole sentences.

4.2 Speech-to-text track

The experimental results are shown in Figure 2. Similarly, the rightmost point is not a result of simultaneous translation. Because the speech in the development set is difficult for ASR, the ASR does not perform well, resulting in a character error rate of 35.3%. The errors caused by ASR are brought to MT, and thus the BLEU is much lower than in the text-to-text track.

4.3 Analysis

From Figure 1 and Figure 2, we can observe that the larger the threshold t is, the longer the latency is. This is because the larger the threshold t is, the longer the detected MUs are, which further leads to longer waiting time between two translations. We can also observe that the larger the threshold t is, the higher the translation quality is. This is because the larger the threshold t is, the more likely a detected MU is a true MU and thus the translation of the detected MU will not be changed by subsequent incoming text. Table 3 is an illustration for this.

Source:	好 , 让 我们 来看 下个 例子 。 okay let we look next example
Reference:	Okay , let 's look at the next example .
Simultaneous Translation ($t = 0.7$):	OK , let 's look at the next example .
Simultaneous Translation ($t = 0.5$):	Okay , || let 's look at the next example .
Simultaneous Translation ($t = 0.3$):	Okay , || let 's do it . || let 's look at the next example .

Table 3: An illustration of text-to-text simultaneous translation with different threshold t. "||" is a symbol for separating the translations of detected MUs.

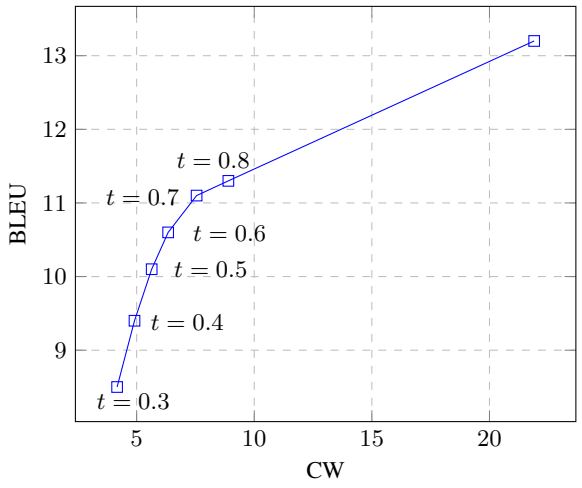

Figure 2: Translation quality against latency of different thresholds t. The rightmost point is not a result of simultaneous translation, but a result got by translating complete sentences.

5 Conclusion

We have built two systems for text-to-text simultaneous translation and speech-to-text simultaneous translation. The key of our systems is the basic adaptive segmentation policy in Zhang et al. (2020). With this policy, simultaneous translation can be achieved without any modification to the MT component, and the latency can be controlled.

In our future work, we would like to study how to improve the cooperation of ASR and MT and the performance of MU classification.

References

N. Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, R. Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *ACL*.

Y. Chen and X. Shi. 2020. Improving machine simultaneous interpretation by punctuation recovery. *Journal of Computer Applications*, 40(4):972–977.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Xutai Ma, J. Pino, James L. Cross, Liezl Puzon, and Jiatao Gu. 2020a. Monotonic multihead attention. *ArXiv*, abs/1909.12406.

Xutai Ma, J. Pino, and Philipp Koehn. 2020b. Simulmt to simulst: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *AACL/IJCNLP*.

Xutai Ma, Yongqiang Wang, M. Dousti, Philipp Koehn, and J. Pino. 2020c. Streaming simultaneous speech translation with augmented memory transformer. *ArXiv*, abs/2011.00033.

Graham Neubig, Kyunghyun Cho, Jiatao Gu, and Victor O. K. Li. 2017. Learning to translate in real-time with neural machine translation. In *EACL*.

J. Niehues, N. Pham, Thanh-Le Ha, Matthias Sperber, and Alexander H. Waibel. 2018. Low-latency neural speech translation. In *INTERSPEECH*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, S. Gross, Nathan Ng, David Grangier, and M. Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL-HLT*.

Kishore Papineni, S. Roukos, T. Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Yi Ren, J. Liu, Xu Tan, C. Zhang, Tao Qin, Zhou Zhao, and T. Liu. 2020. Simulspeech: End-to-end simultaneous speech to text translation. In *ACL*.

Rico Sennrich, B. Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. *ArXiv*, abs/1508.07909.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Ruiqing Zhang, Xiyang Wang, Chuanqiang Zhang, Zhongjun He, Hua Wu, Zhi Li, Haifeng Wang, Ying Chen, and Qinfei Li. 2021. Bstc: A large-scale chinese-english speech translation dataset. *arXiv preprint arXiv:2104.03575*.

RuiQing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2020. Learning adaptive segmentation policy for simultaneous translation. In *EMNLP*.