BIT's system for the AutoSimTrans 2020

Minqin Li, Haodong Cheng, Yuanjie Wang, Sijia Zhang
Liting Wu, and **Yuhang Guo***
Beijing Institute of Technology, Beijing, China
lmqminqinli@163.com guoyuhang@bit.edu.cn

Abstract

This paper describes our machine translation systems for the streaming Chinese-to-English translation task of AutoSimTrans 2020. We present a sentence length based method and a sentence boundary detection model based method for the streaming input segmentation. Experimental results of the transcription and the ASR output translation on the development data sets show that the translation system with the detection model based method outperforms the one with the length based method in BLEU score by 1.19 and 0.99 respectively under similar or better latency.

1 Introduction

Automatic simultaneous machine translation is a useful technique in many speech translation scenarios. Compared with traditional machine translations, simultaneous translation focuses on processing streaming inputs of spoken language and achieving low latency translations. Two challenges have to be faced in this task. On one hand, few parallel corpora in spoken language domain are open available, which leads to the fact that the translation performance is not as good as in general domain. On the other hand, traditional machine translation takes a full sentence as input so that the latency of the translation is relatively long.

To deal with the shortage of the spoken language corpora, we pre-train a machine translation model on general domain corpus and then fine-tune this model with limited spoken language corpora. We also augment the spoken language corpora with different strategies to increase the in-domain corpora.

In order to reduce the translation latency, we use three sentence segmentation methods: a punctuation based method, a length based method and a sentence boundary detection model based method. All of the methods can split the input source sentence into short pieces, which makes the translation model obtain low latency translations.

In the streaming automatic speech recognition(ASR) output track for the Chinese-to-English translation task of AutoSimTrans 2020, most of our proposed systems outperform the baseline systems in BLEU score and the sentence boundary detection model based sentence segmentation method abstains higher BLEU score than the length based method under similar latency.

2 Task Description

We participated in the streaming Chinese-to-English translation task of AutoSimTrans 2020 [1]: the streaming ASR output translation track and the streaming transcription translation track. The two tracks are similar except that the ASR output may contain error results and includes no internal punctuation but end punctuation. Table 1 shows an example of the streaming ASR output translation.

3 Approaches

Our all systems can be divided into 3 parts: data preprocessing, sentence segmentation and translation. Data preprocessing includes data cleaning, data augmentation. We implement 3 sentence segmentation methods, which are based on punctuation, sentence length and a sentence boundary detection model. The training of translation model includes pretraining out of domain and fine-tuning in domain.

*Corresponding author.

[1] https://autosimtrans.github.io/shared

Proceedings of the 1st Workshop on Automatic Simultaneous Translation, pages 37–44
July 10, 2020.

Streaming ASR output	Translation
大	
大家	
大家好	
大家好欢迎	Hello everyone.
大家好欢迎大	Welcome
大家好欢迎大家	
大家好欢迎大家来到	everyone
大家好欢迎大家来到这里	to come
大家好欢迎大家来到这里,	here.

Table 1: An example of streaming ASR output translations.

3.1 Data Cleaning

Noises in large-scale parallel corpus are almost inevitable. We clean the parallel corpus for the training. Here we mainly focus on the miss-aligned errors in the training corpus. We find that in the CWMT19 zh-en data set, some of the target sentences are not in English, but in Chinese, Japanese, French or some other noisy form. We suspect these small noises may affect the training of the model. Inspired by Bérard et al. (2019), we apply a language detection script, *langid.py*[2], to the source and the target sentence of the CWMT19 data set separately. Sentence pairs which are not matched with their expected languages are deleted. The corpus are then cleaned by the *tensor2tensor*[3] module by default. Eventually the CWMT19 corpus are then filtered from 9,023,708 pairs into 7,227,510 pairs after data cleaning.

3.2 Data Augmentation

Insufficiency of training data is common in spoken language translation, and many data augmentation methods are used to alleviate this problem (Li et al., 2018). In the streaming ASR output translation system, we use the homophone substitution method to augment the training data according to the characteristics of ASR output translation. The results of ASR usually contain errors of homophonic substitution. We randomly replace each character in the source language part of the training corpus with probability p with its homophones to improve the generalization ability of the system. As shown in Table 2, we find characters that are homophonic with the selected characters, sample them according to the probability that these characters appear in the corpus, and substitute them to the corresponding positions. The data augmentation is only used in our MT model's training because of the insufficiency of training data in spoken language domain.

Similarly, we randomly substitute words in the source language sentences with the homophone substitution. The result of this substitution is closer to the real speech recognition result. As shown in Table 3. We first split the sentence in the source language into a word sequence, determine whether to replace each word with its homophones by probability p, and then sample them according to the distribution of homophones in a corpus. Finally we replace to the corresponding position.

In this system, we adopt the character and the word frequency distribution in an ASR corpus, the AISHELL-2 corpus (Du et al., 2018), and set the substitution probability $p = 0.3$.

3.3 Sentence Segmentation

Low latency is important to simultaneous machine translation. Our systems are closed to low latency translation by splitting long input word sequences into short ones. We use three sentence segmentation methods in this work, namely, punctuation based sentence segmentation (PSS), length based sentence segmentation (LSS), and sentence boundary detection model based sentence segmentation (MSS).

PSS In the punctuation based sentence segmentation method we put the streaming input tokens into a buffer one by one. When the input token is a punctuation, the word sequence in the buffer is translated. Then the buffer is cleared and we put the next tokens into it. The above procedure repeats until the end of the streaming inputs.

LSS In our length based sentence segmentation method we put the steaming input tokens into a buffer one by one. When the input token is a punctuation or the sequence length in the buffer reaches a threshold L, the word sequence in the buffer except the last word is translated in case of the last word is an in complete one. The translated part in the buffer is then cleared and then we put the next tokens

[2]https://github.com/saffsd/langid.py
[3]https://github.com/tensorflow/tensor2tensor

Original Chinese	这个	**社** (she)	会	没有	信任	没法	运转
English	This	society	society	hasn't	trust	it doesn't	work
Substitution	这个	**设** (she)	会	没有	新人	没法	运转
English	This	suppose	society	hasn't	newcomers	it doesn't	work

Table 2: A randomly selected single character (in red bold font) is substituted by its homophonic character. The corresponding pinyin is included in the bracket.

Original Chinese	这个	社会	没有	**信任** (xinren)	没法	运转
English	This	society	hasn't	trust	it doesn't	work
Substitution	这个	社会	没有	**新人** (xinren)	没法	运转
English	This	society	hasn't	newcomers	it doesn't	work

Table 3: A randomly selected word (in red bold font) is substituted by its homophonic word. The corresponding pinyin is included in the bracket.

into the buffer. The above procedure repeats until the end of the streaming inputs.

Text	**Label**
所以我们认为免费 So we think that free	0
所以我们认为免费只是暂时的 So we think that free is only temporary	1

Table 4: Examples of the train data set of the model. 1: Complete sentences. 0: Incomplete sentence.

MSS Apparently many translation inputs with the LSS are incomplete sentences fragments because of the "hard" sentence segmentation. Here we propose a sentence boundary detection model for the sentence segmentation. We build this model on the top of a pre-training model, BERT(Devlin et al., 2018). Our model is built by adding two layers of full connected network to the Chinese BERT pre-training model. The training data set is constructed using all transcription pairs provided by the organizer. For the sentences in transcriptions, we use a punctuation set, {, . ! ? }, as the sentence boundary indicators to obtain complete sentences, which are used as positive samples. And then we sample incomplete fragments from the above sentences uniformly to obtain negative samples. The ratio of the positive sample to the negative sample is 1 : 4. Table 4 illustrates a positive example and a negative example. The training set is of 370k examples, the test set is of 7k examples, and the validation set is of 7k examples. After running 3 epochs, the model converges with an accuracy of 92.5% in the test set.

We apply the sentence boundary detection model to streaming ASR output translation. The model returns the prediction to each streaming sequence as a judgment condition for whether it is to be translated. However, we should not set the segmentation point at the first position of the detection. Suppose a detected sentence boundary position is i and the next detected boundary position is $i+1$. This means both of the prefix word sequences $w_{1:i}$ and $w_{1:i+1}$ can be seen as a complete sentence. Usually the boundary position $i+1$ is better than i. Generally we set a rule that position i is a sentence boundary if the sentence boundary detection model returns true for position i and false for $i+1$. In this way, the word sequence (i.e. $w_{1:i}$) is feed to the translation system when it is detected and the untranslated part (i.e. w_{i+1}) will be translated in the next sentence. For example, the position i of streaming inputs in Table 5 are detected to boundary's position finally only when the position i is detected to boundary by model while the next position $i+1$ isn't detected to boundary by model.

3.4 Pre-training and Fine-tuning

Pre-training and fine-tuning are the most popular training methods in the field of deep learning. It has been proved that this training mode is very effective in improving the performance of the model and is very simple to implement. Therefore, we use the CWMT19

Position	Sentence	Return of model	Boundary
$i-2$	她喜欢那个公司的设	False	0
$i-1$	她喜欢那个公司的设计	True	0
i	她喜欢那个公司的设计师	True	1
$i+1$	她喜欢那个公司的设计师因	False	0

Table 5: The examples of using model to detect boundaries. 0: Not boundary of sentence, 1: Boundary of sentence

data set to pre-train a base-model, and then use the speech translation data provided by the organizer to fine-tune the model.

We first train a basic Transformer translation model with CWMT19 data set. In order to adapt to the spoken language domain, we directly fine-tune the pre-trained model on the transcriptions or ASR outputs provided by the organizer and our augmented data.

4 Experiments

4.1 Data Sets

Data Set	# Sentence Pairs
CWMT19	9,023,708
Transcriptions	37,901
ASR Outputs	202,237
Development Set	956

Table 6: The size of different data sets.

We train our model with the CWMT19 zh-en data set, the streaming transcription and the streaming ASR output data sets provided by the evaluation organizer. Because of the evaluation track limit, we did not use the UN parallel corpus and the News Commentary corpus although they were used in the baseline. The CWMT19 zh-en data set includes six sub data sets: the casia2015 corpus, the casict2011 corpus, the casict2015 corpus, the datum2015 corpus, the datum2017 corpus and the neu2017 corpus. The CWMT19 data set contains totally 9,023,708 parallel sentences. They are used in the pre-training of our model. Streaming transcription and streaming ASR output data sets are provided by the evaluation organizer. The transcription data set contains 37,901 pairs and the ASR output data set contains 202,237 pairs. We use them as the fine-tuning data to adapt to the spoken language. Finally we evaluate our system on the development set which contains 956 pairs. The size of the data set is listed in Table 6.

4.2 System Settings

Our model is based on the transformer in *tensor2tensor*. We set the parameters of the model as $transformer\_big$. And we set the parameter *problem* as $translate\_enzh\_wmt32k\_rev$. We train the model on 6 RTX-Titan GPUs for 9 days. Then we use the transcription data and the ASR output data to fine-tune the model respectively on 2 GPUs. We fine-tune the model until it overfits.

4.3 Baseline Model

The baseline model[4] (Ma et al., 2018) provided by the evaluation organizer is trained on the WMT18 zh-en data set, including CWMT19, the UN parallel corpus, and the News Commentary corpus. The baseline model uses the transformer which is essentially the same as the base model from the original paper (Vaswani et al., 2017). It applied a Prefix-to-Prefix architecture and Wait-K strategy to the transformer. We test the Wait-1, Wait-3 and the FULL model with fine-tuning on domain data as the comparison to our system. For the Wait-1, Wait-3 setting, the baseline fine-tunes 30,000 steps. For the FULL setting, the baseline fine-tunes 40,000 steps.

4.4 Latency Metric: Average Lagging

Ma et al. (2018) uses Average Lagging (AL) as the latency metric. They defined:

$$\mathrm{AL}_g(\mathbf{x}, \mathbf{y}) = \frac{1}{\tau_g(|\mathbf{x}|)} \sum_{t=1}^{\tau_g(|\mathbf{x}|)} g(t) - \frac{t-1}{r} \quad (1)$$

Where $\tau_g(|\mathbf{x}|)$ denotes the cut-off step which is the decoding step when source sentence finishes, $g(t)$ denotes the number of source words

[4]https://github.com/autosimtrans/SimulTransBaseline

processed by the encoder when deciding the target word $\mathbf{y}_t$, and $r = |x|/|y|$ is the target-to-source length ratio. The lower the AL value, the lower the delay, the better the real-time system.

5 Results

5.1 Streaming Transcription Translation

The results of our streaming transcription system on the development data set are shown in Table 7. FT-Trans indicates the fine-tuning data set including the original transcriptions and the transcriptions without punctuation (i.e. the depunctuation version). LSS-L indicates the system with the length based sentence segmentation method and the threshold for the length is L. PSS indicates the system with our punctuation based sentence segmentation method. MSS indicates the system with our sentence boundary detection model based sentence segmentation method. Wait-1, Wait-3 and FULL indicate the different settings of the baseline systems. Among these settings, the best AL score is from the Wait-1 baseline and the best BLEU score is from our PSS system. Under similar BLEU score, LSS-17 obtains better AL score than the FULL baseline. Both of the AL and the BLEU score of the LSS-L system grow up with L increases. The MSS system performs better BLEU score by 1.19 than the LSS-L system under similar AL score (i.e. MSS vs. LSS-12).

Finally we submitted the PSS setting system because of its high BLEU score and relatively low AL latency compared with the FULL baseline.

Models	Settings	AL	BLEU
FT-Trans	LSS-10	5.9273	17.31
	LSS-11	6.3180	18.12
	LSS-12	6.6932	18.36
	LSS-15	7.7651	20.71
	LSS-17	8.2813	21.84
	PSS	**10.0667**	**24.23**
	MSS	6.7249	19.55
Baseline	Wait-1	2.1014	15.07
	Wait-3	5.1424	17.95
	FULL	24.9331	21.65

Table 7: The translation results on the development data set of streaming transcriptions.

5.2 Streaming ASR Output Translation

The translation performances on the streaming ASR output are shown in Table 8. FT-ASR represents the systems are fine-tuned on the combination of the ASR output and the ASR output without punctuation. FT-ASR+Aug represents the fine-tuning set includes the FT-ASR, the homophone substitution augmented transcriptions, and their depunctuation version. FT-ASR+Aug+Trans represents the fine-tuning set contains the FT-ASR+Aug and the transcriptions and their depunctuation version.

As shown in Table 8, all of our systems outperform the Wait-1, Wait-3 settings of the baseline in BLEU score and our MSS model outperforms the FULL baseline. As more data is added to the fine-tuning set, the performances of the systems will increase accordingly. Both LSS-15 and PSS in FT-ASR+Aug outperform the corresponding systems in FT-ASR, which indicates the effectiveness of the data augmentation. The BLEU score of LSS-15(FT-ASR+Aug+Trans) is 2.22 higher than LSS-15(FT-ASR) while the AL latency of former is better than the latter.

In the FT-ASR+Aug+Trans, the sentence boundary detection model based sentence segmentation, MSS, obtains higher (i.e. +0.99) BLEU score and lower (i.e. -1.06) AL latency than the LSS-15. The BLEU score of MSS is lower than PSS by 1.46 but the latency is improved by 15.88.

Compared with the results of transcription translation of FT-Trans in Table 7, the BLEU scores of the ASR outputs translations relatively decreased. This indicates the effects of the cascade error of the ASR systems.

The latency of the LSS in Table 7 and Table 8 are close. The latency of PSS increased from 10 to around 22. This indicates the lack of punctuation in the ASR outputs.

The MSS system performs close AL latency and less BLEU score drops in transcription and ASR outputs translation. At last we submitted the MSS system to the evaluation track.

Several examples of the translation in differ-

Models	Settings	AL	BLEU
FT-ASR	LSS-15	7.5636	13.62
	PSS	22.0051	18.23
FT-ASR	LSS-15	7.2222	14.99
+Aug	PSS	21.9600	18.29
FT-ASR	LSS-15	7.1298	15.84
+Aug	PSS	21.9557	18.29
+Trans	**MSS**	**6.0709**	**16.83**
Baseline	Wait-1	1.0766	10.72
	Wait-3	3.9692	12.75
	FULL	24.0415	15.13

Table 8: The translation results on the development data set of streaming ASR outputs.

ent systems can be seen in Appendix A.

6 Related Work

End-to-end machine translation models, such as transformer (Vaswani et al., 2017), greatly promote the progress of machine translation research and have been applied to speech translation researches (Schneider and Waibel, 2019; Srinivasan et al., 2019; Wetesko et al., 2019). Furthermore, several end-to-end based approaches have recently been proposed for simultaneous translations (Zheng et al., 2019b,a).

In order to solve the problem of insufficient parallel corpus data for simultaneous translation tasks, Schneider and Waibel (2019) augmented the available training data using back-translation. Vial et al. (2019) used BERT pre-training model to train a large number of external monolingual data to achieve data augmentation. Li et al. (2018) simulated the input noise of ASR model and used placeholders, homophones and high-frequency words to replace the original parallel corpus at the character level. Inspired by Li et al. (2018), we augment the training data by randomly replacing the words in the source sentences with homophones.

In order to reduce the translation latency, Ma et al. (2018) used the Prefix-to-Prefix architecture, which predicts the target word with the prefix rather than the whole sequence. Their Wait-K models are used as the baseline and are provided by the shared task organizers. The Wait-K models start to predict the target after the first K source words appear. Zheng et al. (2020) applied ensemble of models trained with a set of Wait-K polices to achieve an adaptive policy. Xiong et al. (2019) have proposed a pre-training based segmentation method which is similar to MSS. However, in the decoding stage, the time complex of this method is $O(n^2)$ whereas the time complex of MSS is $O(n)$.

7 Conclusions

In this paper, we describe our submission systems to the the streaming Chinese-to-English translation task of AutoSimTrans 2020. In this system the translation model is trained on the CWMT19 data set with the transformer modedalvi2018incrementall. We leverage homophonic character and word substitutions to augment the fine-tuning speech transcription data set. We implement a punctuation based, a length based and a sentence boundary detection model based sentence segmentation methods to improve the latency of the translation system. Experimental results on the development data sets show that the punctuation based sentence segmentation obtains the best BLEU score with a reasonable latency on the transcription translation track. The results on the ASR outputs translation show the effectiveness of our data augmentation approaches. And the sentence boundary detection model based sentence segmentation gives the low latency and a stable BLEU score in our all systems. However, because we have no enough time to retrain the MT model, some settings of our system are not consistent with the baseline, so it is difficult to judge whether our method is better than baseline's method. In the future, we will finish this comparative experiment.

Acknowledgments

Supported by the National Key Research and Development Program of China (No. 2016YFB0801200)

References

Alexandre Bérard, Ioan Calapodescu, and Claude Roux. 2019. Naver labs europe's systems for the wmt19 machine translation robustness task. *arXiv preprint arXiv:1907.06488*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu. 2018. AISHELL-2: transforming mandarin ASR research into industrial scale. *CoRR*, abs/1808.10583.

Xiang Li, Haiyang Xue, Wei Chen, Yang Liu, Yang Feng, and Qun Liu. 2018. Improving the robustness of speech translation. *arXiv preprint arXiv:1811.00728*.

Mingbo Ma, Liang Huang, Hao Xiong, Kaibo Liu, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, and Haifeng Wang. 2018. STACL: simultaneous translation with integrated anticipation and controllable latency. *CoRR*, abs/1810.08398.

Felix Schneider and Alex Waibel. 2019. Kit's submission to the iwslt 2019 shared task on text translation. In *Proceedings of the 16th International Workshop on Spoken Language Translation*.

Tejas Srinivasan, Ramon Sanabria, and Florian Metze. 2019. Cmu's machine translation system for iwslt 2019.

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

Loïc Vial, Benjamin Lecouteux, Didier Schwab, Hang Le, and Laurent Besacier. 2019. The lig system for the english-czech text translation task of iwslt 2019.

Joanna Wetesko, Marcin Chochowski, Pawel Przybysz, Philip Williams, Roman Grundkiewicz, Rico Sennrich, Barry Haddow, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. Samsung and university of edinburgh's system for the iwslt 2019.

Hao Xiong, Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Dutongchuan: Context-aware translation model for simultaneous interpreting. *arXiv preprint arXiv:1907.12984*.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. 2020. Simultaneous translation policies: From fixed to adaptive.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019a. Simpler and faster learning of adaptive policies for simultaneous translation. *arXiv preprint arXiv:1909.01559*.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019b. Simultaneous translation with flexible policy via restricted imitation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5816–5822, Florence, Italy. Association for Computational Linguistics.

A Appendices

We list several translation results to compare our systems with the baselines on the transcription translation track and the ASR output translation track. As shown in Table 9 and 10, missing translation can be observed in the Wait-K baselines and our system.

Source	**Reference**
在他每一次比赛都是输，甚至是倒数第一第二名的时候，是什么，是什么样的力量支撑着他一直去比赛，一直去训练。	He has always been ranked among the last, so to speak, the last in those games. What kind of spirit supported him to take part in the competition all the time?

Table 9: An example of source sentence and reference translation in the transcription translation track.

For streaming ASR output, as shown in Table 12, missing translation can also be observed in the Wait-K baselines. From Table 13, we can see that in the segmentation of the LSS-15 most of the sentence fragments are incomplete. As shown in Table 14, the segmentation of the MSS is reasonable and the translation is much better than the LSS-15.

System	Translation
Wait-1	In his every after shock, he won the game, even in the No.1 games.
Wait-3	Every time when he does a match, he will lose, even in the No.1 draw, what is that?
FULL	In every game, which is not only about the win, but also about the power that comes to the 1st place, those who support him to go on training all the time.
FT-Trans(PSS)	In every game he lost, in the second countdown, what is it? What was the strength that kept him going? I keep training.

Table 10: The translations of the sentence in Table 9.

Source	Reference
对吗每个人都是不想输的都是想赢的在它每一次比赛都是输甚至是倒数第第二名的时候什么是什么样的力量支撑着他一直去比赛	Right? Everyone does not want to lose; rather, they all want to win.When he lost every match or even came in the second last or last place, what was it or what kind of strength supported him to compete and train all the time?

Table 11: An example of the source sentence and the reference translation in the ASR output translation track.

System	Translation
Wait-1	So, is everyone wants to fail?
Wait-3	Right, everyone never want to fail, and they all want to win every game, even when they are in the second best.
FULL	That is, to say, every one would never want to win, in every game, or even in the second place, what was the power that supports him to go there and that number?

Table 12: The translations of the sentence in Table 11.

Segmentation	Translation
对吗每个人都是不想输的都是想	Yes, everyone wants to lose.
赢的在它每一次比赛都是输甚至	The winner lost every game.
是倒数第第二名的时候什么是	What is second to last?
什么样的力量支撑着他一直去	What kind of strength supports him to go on?
比赛	The game.

Table 13: The sentence segmentation and the corresponding translations in Table 11 with the setting of LSS-15 on FT-ASR+Aug+Trans.

Segmentation	Translation
对吗每个人都是不想输的	Right? Everyone doesn't want to lose.
都是想赢的	They all want to win.
在它每一次比赛都是输甚至是倒数	In each game, it is losing or even losing.
第第二名的时候	In the second place.
什么是什么样的力量	What is power?
支撑着他一直去比赛	It supports him to go all the way to the game.

Table 14: The sentence segmentation and the corresponding translations in Table 11 with the setting of MSS on FT-ASR+Aug+Trans.