ISTIC's Neural Machine Translation System for IWSLT 2020

Jiaze Wei, Wenbin Liu, Zhenfeng Wu, You Pan, Yanqing He[1]
Institute of Scientific and Technical Information of China
No 15, Fuxing Road, Haidian District,Beijing, China, 100038
{weijz2018, liuwb2019, wuzf, pany, heyq}@istic.ac.cn

Abstract

This paper introduces technical details of machine translation system of Institute of Scientific and Technical Information of China (ISTIC) for the 17th International Conference on Spoken Language Translation (IWSLT 2020). ISTIC participated in both translation tasks of the Open Domain Translation track: Japanese-to-Chinese MT task and Chinese-to-Japanese MT task. The paper mainly elaborates on the model framework, data preprocessing methods and decoding strategies adopted in our system. In addition, the system performance on the development set are given under different settings.

1 Introduction

This paper describes the neural machine translation (NMT) system of the Institute of Scientific and Technical Information of China (ISTIC) for the 17th International Conference on Spoken Language Translation (IWSLT 2020) (Ebrahim et al., 2020). ISTIC participated in the Japanese-to-Chinese and Chinese-to-Japanese MT tasks of the Open Domain Translation track.

In this evaluation, we adopted the NMT Google Transformer (Vaswani et al., 2017) architecture as a part of our system. We use the data released by the organizer and adopted general and specific preprocessing methods to the training and development data. Several filtering methods of corpus are explored to improve the quality of the training data. A corpus filtering method based on Elasticsearch is used to select the development data similar to test data. We adopted a model averaging strategy in the decoding phase and different results are combined in post-processing stage to obtain the final translation. The performance of the system is compared under different settings in the two translation directions, and further analyzed.

2 System Architecture

Figure 1 shows the flow chart of ISTIC's NMT system in this evaluation. Our model architecture, data processing and decoding strategy are given below.

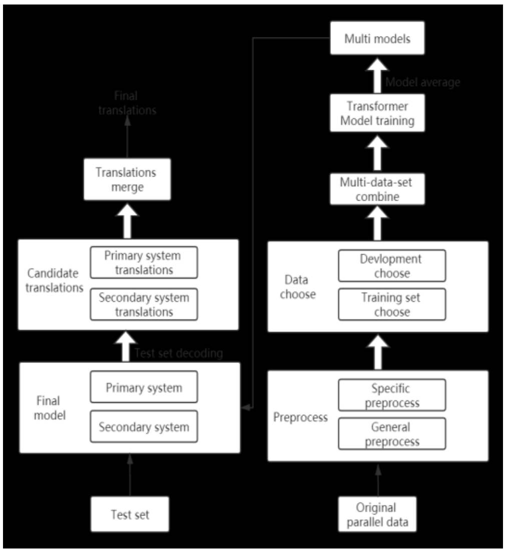

Figure 1. Overall flow chart of evaluation

[1] Responsible author, heyq@istic.ac.cn. This research has been partially supported by ISTIC research foundation projects ZD2019-20, ZD2020-10.

Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT), pages 158–165
July 9-10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

Our baseline system used in this evaluation is the Transformer (Vaswani et al., 2017) based on a full attention mechanism, which includes an encoder and a decoder, as shown in Figure 2. Transformer does not use a recurrent neural network (Cho et al., 2014) or a convolutional neural network (Gehring et al., 2017), but is completely based on attention mechanism. It can achieve algorithm parallelism, speed up model training, further alleviate long-distance dependence and improve translation quality.

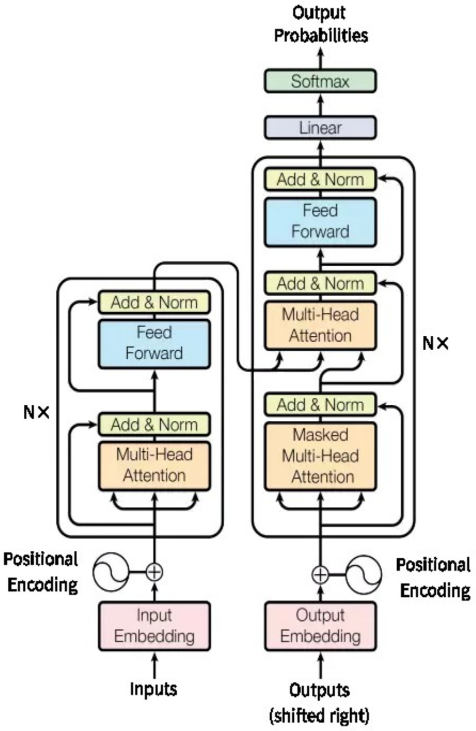

Figure 2. Transformer model (Vaswani et al., 2017)

The encoder and decoder are formed by stacking N layer blocks. Each layer of encoder contains two sub-modules, namely a multi-head self-attention module and a feed-forward neural network module. The multi-head self-attention module divides the dimension of hidden state into multiple parts， and each part is separately calculated by using self-attention function, furthermore, these output vectors are concatenated together. Multi-head mechanism enables the model to pay more attention to the feature information of different positions and different sub-spaces. The multi-head attention method includes two steps: 1) dot product attention calculation; 2) multi-head attention calculation. The calculation method of dot product attention can be expressed as:

$$\mathrm{Attention}(Q,K,V)=\mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

where Q is the query vector, K is the key vector, V is the value vector, and d_k is the dimension of the hidden layer state. On the basis of dot product attention, the calculation method of the multi-head attention mechanism can be expressed as:

$$\mathrm{MultiHead}(Q,K,V)=\mathrm{Concat}(\mathrm{head}_1,\ldots,\mathrm{head}_h)W^O$$

where W^0 is the matrix parameter. The attention value of each head is:

$$\mathrm{head}=\mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Each layer of the decoder is composed of three sub-modules. In addition to the two modules similar to the encoder, an decoder-encoder attention module is added between them and can focus attention on source language information in decoding process. In order to avoid the problem that too many layers cause the model to be difficult to converge, both the encoder and the decoder use residual connection and hierarchical regularization techniques. To make the model obtain the position information of the input sentence, additional position encoding vectors are added to the input layer of the encoder and decoder.

After the encoder obtains a hidden state, Transformer model inputs the hidden state into the softmax layer and scores with candidate vocabulary to obtain the final translation result.

3 Data processing

3.1 Corpus preprocessing

In this evaluation we use data of the open domain translation track released by the evaluation organizer of IWSLT 2020, as shown in Table 1. It contains existing

Data1	web_crawled_parallel_filtered.tar.gz
Data2	existing_parallel.tar.gz
Data3	web_crawled_parallel_unfiltered.tar.gz
Data4	web_crawled_unaligned.tar.gz

Table 1. Data provided by reviewer

Japanese-Chinese parallel data (Data 2) and web crawl data (Data 1, Data 3, Data 4)

The quality of Data 2 is much better than other web crawling data. Therefore, a two-stage preprocessing method is designed as a general preprocessing stage and a specific preprocessing stage.

General preprocessing stage: Due to time limitation, we did not have a chance to use Data 4, only the following preprocessing operations were performed on Data 1, Data 2 and Data 3:

- Traditional Chinese to Simplified Chinese
- Word segmentation
- Filtering of adjacent similar sentences
- Sentence length filtering, sentence length ratio filtering
- Language token ratio filtering
- Special character filtering

Among them, the filtering of adjacent similar sentences calculates the Dice similarity (Dice, 1945) of the current sentence with the previous sentence in the corpus of source language side or target language side, and remove the current sentence pair if the Dice similarity exceeds 0.9. Sentence length filtering removes sentence pairs which source sentence length or target sentence length is 0 or exceeds 50, and sentence length ratio filtering excludes the sentence pairs whose ratio of source sentence length and target sentence length exceeds the range of [0.2, 5]. Since a certain percentage of sentence pairs in the corpus use the same language as the source and target sentences. The language token ratio method (Lu et al., 2018) is used to eliminate sentence pairs where the proportion of Japanese or Chinese words is smaller than a certain threshold， here set to 0.1. Both Japanese and Chinese word segmentation are implemented using the lexical tool Urheen[2].

Specific preprocessing stage: The quality of training corpus has a great influence on the performance of machine translation model. The web crawling data is large in scale but has a great amount of noise, thus, the following specific preprocessing operations are performed on Data 1 and Data 3:

GIZA++[3] tool is used on Data 2 to obtain an alignment dictionary, and each word only retains the top ten translations in their probability ranking. According to the alignment dictionary, the alignment scores for each sentence pair in Data 1 and Data 3 are calculated and the threshold is set as 0.4 as the alignment ratio:

$$alignment_{ratio(X,Y)} = \frac{\sum_x \sum_y p(x,y)}{length(X)} + \frac{\sum_y \sum_x p(y,x)}{length(Y)}$$

where X is a source sentence, Y is a target sentence; $alignment_{ratio(X,Y)}$ is the alignment ratio of sentence pair(X,Y); x is the word in sentence X, y is the word in sentence Y; p(x, y) is the probabilities that word x translates into word y, p(y, x) is the probabilities that word y translates into word x; and $length(X)$ is the length of sentence X; $length(Y)$ is the length of sentence Y.

The filtering results after general preprocessing and specific preprocessing are shown in Table 2.

Data	Original sents number	After filtering
Data1	18966595	8531325
Data2	1963238	1726668
Data3	160M+	44557281

Table 2. Data filtering results

3.2 Elasticsearch similar corpus filtering

In order to further improve the consistency of the development set and test set， and further

[2] https://www.nlpr.ia.ac.cn/cip/software.html

[3] https://github.com/moses-smt/giza-pp

optimize the machine translation performance, we choose similar sentence pairs from Data 1 and Data 2 for each sentence in test set to build a new development set and a new test set. We define them as ES development set, ES test set. ES development set is used for early stopping and fit the model to the test set. ES test set is used to compare the performance of the original development set and ES development set. Specifically, for $q_i, 1 \le i \le 875$, in each test set sentence, we create an index base D of Data 1 and Data 2 to retrieve similar sentences from D with the Elasticsearch[4] retrieval tool (version number: v6.1.0). Elasticsearch returns a similarity score between q_i and each sentence $d_j, 1 \le j \le |D|$ in D:

$$score(q_i,d_j) = coord(q_i,d_j) * queryNorm(q_i) * \sum_{t\ in\ q_i} \left(tf(t\ in\ d_j) * idf(t)^2 * boost(t) * norm(t,d_j) \right)$$

where, $score(q_i,d_j)$ is the similarity score of the test set sentence q_i with the sentence d_j of Data 1 and Data 2; $1 \le i \le 875$, $1 \le j \le |D|$, $|D|$ represents the total number of sentences of Data 1 and Data 2; $coord(q_i,d_j)$ is the coordination factor between sentence q_i and sentence d_j; $queryNorm(q_i)$ is the normalization factor of query sentence q_i; $tf(t\ in\ d_j)$ is the frequency of word t in sentence d_j; $idf(t)$ is the inverse document-word frequency of word t; $boost(t)$ is the weight used to query the word t; $norm(t,d_j)$ is the length norm of the sentence d_j when querying word t. We select rank first and third sentence pairs in similarity in D to build a new development set -- ES development set, and rank second sentence pairs in similarity to a new test set -- ES test set. After filtering out duplicate sentence pairs and low-quality sentence pairs, the development sets and test sets for Zh-Ja and Ja-Zh eventually obtained as shown in Table 3.

The Data	Original	ES	
		Zh - Ja	Ja - Zh
Development set	5304	1609	1557
Test set	875	904	869

Table 3. selection of similar data

4 Decoding strategy

4.1 Model average

In order to reduce model parameter instability and improve model robustness, the model averaging technique was applied on the parameters stored in the same model at different training moments. We average the parameters of the last N epochs when the model is converged. N is set to 5 for this evaluation.

4.2 Candidate translations merge

Data 1 and Data 2 are included in training set. However, Data 1 contains a great amount of noise and results in some untranslated sentences in the Zh-Ja translations task, some of which take the source sentences directly as the target translations. This rarely happens in the translation model which was trained only with Data 2. Therefore, the former system is taken as primary system and the latter as secondary

[4] https://www.elastic.co/guide/cn/elasticsearch/guide /current/practical-scoring-function.html

system. The final translations are obtained by combining two systems' translations. For each source sentence in the test set of the Zh-Ja task, primary system translations and secondary system translations are checked by the following standards: 1) the primary system translation are exactly the same as the source sentence; 2) the primary system translation are judged as non-Japanese words by the language detection tool. If one of the two checks is satisfied and the secondary system translation is also judged to be Japanese, then the secondary system translation will replace the primary system translation as final translation.

5 Experimental results

5.1 Parameters setting

The open source project tensor2tensor[5] is chosen for this evaluation system. The main parameters are set as follows. Each model uses 1-3 GPUs for training, and the batch size is 2048. We use six self-attention layers for both encoder and decoder, and the multi-head self-attention mechanism has 16 heads. The embedding size and hidden size are set to 1024, the dimension of the feed-forward layer is 4096 and ReLU (Krizhevsky et al., 2012) is used as the activation function. The dropout mechanism (Gal and Ghahramani, 2015) was adopted, and dropout probabilities are set to 0.1. BPE (Sennrich et al., 2015) is used in all experiments, where the merge operation is set to 30K. The initial learning rate is 0.2, and the warm-up steps are set to 8000.

To choose the method of word segmentation, we use Data 2 as training data and score on the development set provided by evaluation organizer, as shown in Figures 3 and 4 where the horizontal axis is the different settings of model parameter alpha, and vertical axis as the character-based bleu4. Zh_ja_b and ja_zh_b means Jieba[6] word segmentation in Chinese (Zh) and Mecab[7] word segmentation in Japanese (Ja). Zh_ja_u and ja_zh_u use the lexical tool Urheen (Zh,Ja) in both the two language.

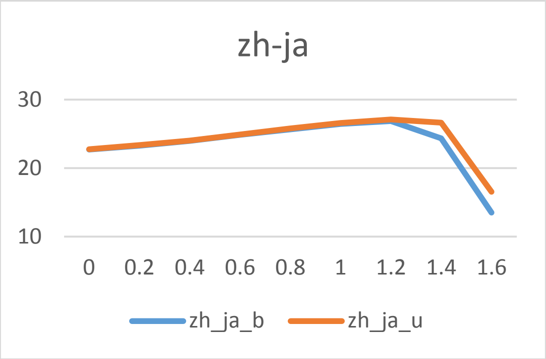

Figure 3. Comparison of word segmentation in Chinese-to-Japanese task

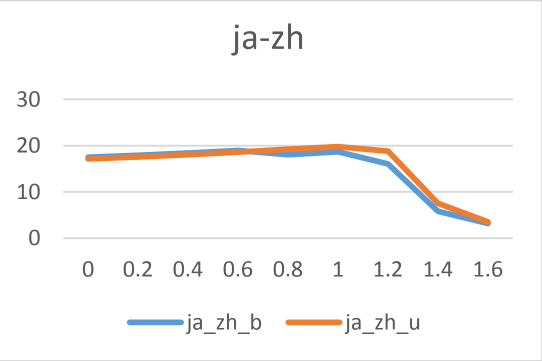

Figure 4. Comparison of word segmentation in Japanese-to-Chinese task

5.2 Experimental results

Training data comparison: In order to choose training data, we use baseline system and score on the original development set, shown in Tables 4-5. It can be seen that increasing the scale of training corpus helps to improve the translation ability. Although Bleu (token) score decreased after Data 1 was added to Chinese-to-Japanese task, we still decided to adopt Data1 + Data2 for training data on the both translation tasks.

[5] https://github.com/tensorflow/tensor2tensor

[6] https://github.com/fxsjy/jieba

[7] https://github.com/SamuraiT/mecab-python3

The direction	Data	Number of sentences	Number of words
Zh-JA	Training data	54M	ZH-613M; JA-753M
	Original + Es Development set	6913	ZH-64K; JA-79K
JA-Zh	Training data	54M	Zh-613M; Ja-753M
	Original + Es Development set	6861	ZH-68K; JA-84K

Table 8. The statistics of data

Data	Bleu (token)	Bleu (character)
Data2	16.76	26.14
Data1 + Data2	15.83	27.01
Data1,2,3	17.64	28.9

Table 4. Training data comparison in Chinese-to-Japanese task

Data	Bleu (token)	Bleu (character)
Data2	11.67	19.95
Data1 + Data2	14.79	26
Data1,2,3	13.97	24.75

Table 5. Training data comparison in Japanese -to-Chinese task

Development data	Bleu (token)	Bleu (character)
Original development set	20.79	28.72
Es development set	21.15	29.6
Original development set + Es development set	21.28	28.96

Table 6. Development set comparison in Chinese-to-Japanese task

Development data	Bleu (token)	Bleu (character)
Original development set	13.91	25.79
Es development set	14.34	25.86
Original development set + Es development set	13.94	25.97

Table 7. Development set comparison in Japanese -to-Chinese task

System	Bleu (token)	Bleu (character)
Data2	18.86	28.82
Data1 + Data2	23.82	35.95
Data1+Data2+ Data3	22.87	35.14
Data2 + avg	19.59	29.81
Data1+Data2+avg	24.51	36.70

Table 9. System comparison inChinese-to-Japan task

System	Bleu (token)	Bleu (character)
Data2	12.00	23.25
Data1 + Data2	20.04	32.09
Data1+Data2+ Data3	16.34	27.72
Data1+Data2+avg	20.98	33.27

Table 10. System comparison in Japanese-to-Chinese task

ES development set: In order to verify the effect of ES development set, we use Data2 as training data, and to train baseline system on different combinations of development sets, and score on ES test set, as shown in Tables 6-7. The experimental results show that ES development set alone or together with the original development set is better than the original development set alone to train machine translation system.

The statistics of training data and development set that are used in this evaluation can be found in Table 8.

System comparison: In order to choose machine translation system, we designed different combinations of training data, and further trying model average strategy on original development set + ES-development set, and score on them, as shown in Tables 9-10. It can be seen that the larger training set Data2+Data1 leads to the result improvement, but no further effect to the growth of training set. Since Data3 was captured from public network, its quality is still limited even though we have

Source-Zh	Data2+avg Translation	Data2+Data1+avg Translation
<sent1>2月13日，日本东京一改前几日的寒凉，迎来一个拨云见日的好天气。(On February 13, Tokyo, Japan ushered in a good weather to see the sun as soon as it changed the cold of the previous days.)	2月13日（2月13日）、日本の東京で少し前の寒さにより、晴れた日を迎える。(February 13 (February 13) greets a sunny day with the recent cold in Tokyo in Japan.)	2月13日，日本东京一改前几日的寒凉，迎来一个拨云见日的好天气。(It's just a copy of Chinese.)
<sent2>阿部知事对中国疫情致以慰问，介绍了长野县为中方抗击疫情提供支持情况，表示当前日本国内正全力阻止疫情扩散，愿同中方继续加强配合，早日战胜疫情。(Abe extended condolences to China on the epidemic and introduced nagano prefecture's support for China's fight against the epidemic. Abe said that Japan is doing its best to prevent the spread of the epidemic and is ready to strengthen cooperation with China to overcome the epidemic at an early date.)	阿部知事は中国の疫病に対する哀悼の意を表明し、長野県を中国側が積極的に支持していることを紹介し、現在の日本国内では流行を阻止しつつあると述べた。(Governor Abe expressed his condolences to the plague of China and introduced the fact that the Chinese side actively supported Nagano Prefecture, and said it was blocking the epidemic in present Japan.)	阿部知事は中国の流行を慰問し、長野県の中で流行に対抗するための支援を紹介し、日本国内では流行の拡散を全面的に阻止していることを明らかにした。(Governor Abe extended condolences to the epidemic in China, introduced support for combating the epidemic in Nagano Prefecture, and made clear that it was completely stopping the spread of the epidemic in Japan.)

Table 11. Translation results merge in Chinses-to-Japanese

adopted a few strategies to filter it. Meanwhile, Data 3 has open domains yet we did not carry out consistency analysis on the data domain. In addition, the model average strategy also brings some improvement to the translation effect.

Therefore, we adopted model average strategy and train on data1+data2 for the two translation tasks. Table 11 shows some translation results of Chinese-to-Japanese translation task. It can be seen that for <sent2>, the translation quality of Data2+Data1+avg model is better than that of Data2+avg model. But due to the noise in Data 1, some translations of the sentences are completely the same as source text. But the situation for Data2+avg model is rare, thus we take the post-processing strategy to merge them. Data1+Data2+avg model is looked as primary system, Data2+avg model as secondary system. Take <sent1> as an example, primary system translation is judged to be in Chinese and is just a copy of Chinese. The secondary translation was checked to be in Japanese, and successfully express the meaning of the source sentence.

Our final system: 1) data filtering; 2) a transformer system trained with Data2+Data1; 3) decoding: model average, candidate translations merge.

6 Conclusions

This paper introduces the main techniques and methods used by the Institute of Scientific and Technical Information of China on the task of two-directions translation of Japanese and Chinese in IWSLT 2020 Open Domain Translation track. We use the architecture of transformer model based on a full attention mechanism. Several filtering methods were explored in data preprocessing, and the model average strategy is adopted in decoding. Similar development set is chosen based on ES and the results are merged by post-processing. Experimental results show that these methods can effectively improve the quality of translation.

Due to limited time, many methods are not able to execute during this evaluation. Our adopted translation model still has a lot room to improve. We expect to learn more advanced techniques and construct a better machine translation system in a short future.

Acknowledgments

We thank the reviewers for their detailed and constructed comments. Yanqing He is the corresponding author. The research work is partially supported by ISTIC research foundation projects ZD2019-20, ZD2020-10.

References

Cho, K., Van Merrienboer, B., Gulcehre, C., et al. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv: Computation and Language.

Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. Ecology, 26(3), 297-302.

Ebrahim Ansari , Amittai Axelrod, Nguyen Bach et al. (2020). Findings of the IWSLT 2020 Evaluation Campaign. Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT 2020), Seattle, USA.

Gal. Y., and Ghahramani. Z. (2015). A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. arXiv: Machine Learning.

Gehring. J., Auli. M., Grangier. D.,et al. (2017). Convolutional Sequence to Sequence Learning. arXiv: Computation and Language.

Krizhevsky. A., Sutskever. I., and Hinton. G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems.

Lu. J., Lv. X., Shi. Y., et al. (2018). Alibaba Submission to the WMT18 Parallel Corpus Filtering Task. Proceedings of the Third Conference on Machine Translation: Shared Task Papers.

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. Neural Information Processing Systems.

Sennrich. R., Haddow. B., and Birch. A. (2015). Neural Machine Translation of Rare Words with Subword Units. arXiv: Computation and Language.