ISTIC's Thai-to-Chinese Neural Machine Translation System for CCMT' 2022

Shuao Guo, Hangcheng Guo, Yanqing He*, and Tian Lan

Research Center of Information Theory and Methodology, Institute of Scientific and Technical Information of China, Beijing, China, 100038
{guosa2021, guohc2020, heyq, lantian}@istic.ac.cn

Abstract. This paper introduces technical details of Thai-to-Chinese neural machine translation system of Institute of Scientific and Technical Information of China (ISTIC) for the 18th China Conference on Machine Translation (CCMT' 2022). ISTIC participated in a low resource evaluation task: Thai-to-Chinese MT task. The paper mainly illuminates its system framework based on Transformer, data preprocessing methods and some strategies adopted in this system. In addition, the paper evaluates the system performance under different methods.

Keywords: Neural Machine Translation · Self-Attention Mechanism · Context-aware System

1 Introduction

ISTIC participated in a low resource evaluation task: Thai-to-Chinese MT task. In this evaluation, our team adopted the Google Transformer architecture as the basis of our system. We collected data from three different sources to form the training set, which were the data released by the evaluation organization, the pseudo parallel corpus and the external data of self-built Thailand-Chinese dictionary and bilingual parallel corpus. The monolingual data released by the evaluation organizer of CCMT' 2021 was filtered to construct the pseudo parallel corpus through the back-translation method, the pseudo parallel corpus and the original given bilingual parallel corpus were used together as the training set of our neural machine translation system. Since the scale of given data was too small, the external data of self-built Thailand-Chinese dictionary and bilingual parallel corpus were introduced into training set. In terms of data pre-processing, we adopted general methods and specific methods for the given data, which mainly included filtering special characters, removing duplicate sentences, and bilingual tokenization. In the construction of the system model, we mainly used the context-aware system method, which took the surrounding sentences as the context and employs an additional neural network to encode the context.We adopted the method of model averaging and ensemble to get the final translation result and removed the spaces between the words of results and finally submitted XML format result to the evaluation organization.

The structure of this paper is as follows: the second part introduces our technical architecture of the machine translation system in this evaluation task; the third part explains the methods used in this evaluation task; the fourth part describes the core process, parameter settings, data pre-processing and experiments results.

2 System Architecture

Figure 1 shows the overall flow chart of our neural machine translation system in this evaluation which includes data pre-processing, data set partition, model training, model inference, and data post-processing.

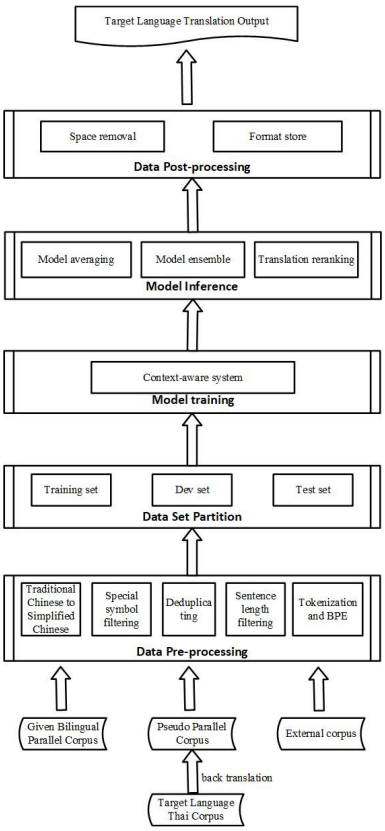

Fig. 1. Overall flow chart for machine translation tasks.

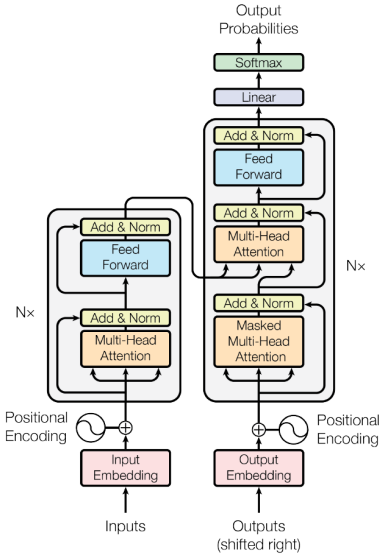

Fig. 2. Transformer model structure

2.1 Baseline System

The baseline system we adopted in this evaluation task is Google's Transformer, which has achieved significant results on machine translation since being proposed in 2017[1]. Its whole network structure is absolutely built on attention mechanism instead of traditional CNN and RNN in deep learning, which has brought a series of advantages, such as consuming less training power, achieving algorithm parallelism, further alleviating long-distance dependence and most importantly, getting a better translation quality. Transformer is essentially an Encoder-Decoder structure, just like most seq2seq models. It consists of Encoder and Decoder(see Fig. 2). Both parts have n stacked identical layer blocks(n can be any number, our system set n to 6.). Every layer of encoder contains two sub-layers(see the left part of Fig. 2), which we call the self-attention sub-layer and the feed-forward sub-layer. The self-attention sub-layer calculates the output representation of a token by attending to all the neighbors in the same layer, computing the correlation score between this token and all the neighbors, and finally linearly combining all the representations of the neighbors and itself.Each layer of decoder includes three parts,masked self-attention mechanism,encoder-decoder attention sub-layer and feed-forward sub-layer[2]. Masked self-attention mechanism is responsible for summarizing the partial prediction history.Encoder-decoder attention sub-layer is used to determine the dynamic source-side contexts for current prediction. A residual connection[3] is employed around each sub-layers in both decoder and encoder, followed by layer normalization[4].

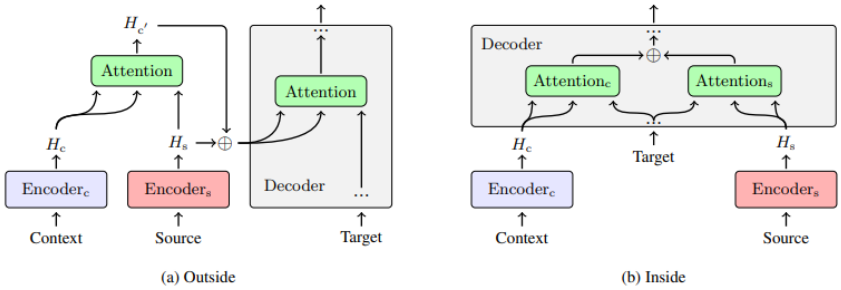

Fig. 3. An overview of two multi-encoder systems. In the Outside approach, *Hs* is the query and *Hc* is the key/value. In the Inside approach, Target is the query, *Hs* and *Hc* represent key/value.

2.2 Our System

Based on the transformer model, we build a context-aware system [5] leaving Transformer's decoder intact while incorporating context information on the

encoder side[6].This approach takes the surrounding sentences as the context and employ an additional neural network to encode the context, that is, there is a source-sentence encoder and a context encoder. Figure 3 shows two methods of integrating the context into NMT.

There are two methods for integrating the context into NMT.The method of outside integration(See Fig.3 (a)) is that the representations of the context and the current sentence are firstly transformed into a new representation by an attention network,then the attention output and the source sentence representation are fused by a gated sum.Alternatively, inside integration(See Fig.3 (b)) means decoder can attend to two encoders respectively and the gating mechanism inside the decoder is employed to obtain the fusion vector.There are two kinds of context that can be used to integrate into NMT.One is source context,another is target context.We often make train set and development set of source language as source context,and make train set and development set of target language as target context.

3 Methods

In this evaluation we try the following methods to improve translation performance.

3.1 Back translation

Back Translation (BT)[6] is one of the most commonly used data augmentation method for machine translation tasks. In our Thai-to-Chinese task, we took three steps to train a Thai-to-Chinese translation model. We train a Chinese-Thai translation model on the released bilingual data and use the model to translate the additional Chinese sentences into Thai sentences as pseudo bilingual sentence pairs, which are mixed with the released sentence pairs to train the final Thai-to-Chinese translation model.

3.2 Add external data

The success of neural machine translation is closely related to computing resources, algorithm models, and data resources, especially the scale of bilingual training data. In the Thai-to-Chinese task, the number of sentence pairs of parallel corpus available for training is as low as 200,000. Therefore, the introduction of external resources can effectively improve the performance of the machine translation system.

3.3 Model averaging

Model averaging[7] refers to averaging the parameters of the same model at different training moments to obtain more robust model parameters, which helps to reduce the instability of model parameters and enhance the robustness of the

model. After specifying the Max EPOCH parameter in the trainer and completing the training process, our team gets the best epoch checkpoint and the last EPOCH checkpoint, and averages the two checkpoints. The more stable and robust individual models obtained through the model averaging strategy will also be used for model averaging to jointly predict probability distributions

3.4 Model ensemble strategy

Model ensemble[8] refers to that in the decoding process, multiple models simultaneously predict the probability distribution of the target word at the current moment, and finally make a weighted average of the probability distribution predicted by multiple models to jointly determine the final output after model ensemble.

4 Experiments

4.1 System Settings

The baseline MT system is based on Transformer trained only by the given bilingual parallel corpus. Outside integration and inside integration are also used in the experiments. Table 1 shows the parameters settings of the three systems. Since context-aware system[9] is fine-tuned on the basis of the baseline system, the value of initial state settings of baseline system is smaller than baseline system. Table 2 shows the Initial learning rate setting of three systems.

Table 1. Fundamental parameters settings of three systems.

Parameter	Value
GPU numberused for each model training	1-3
batch size	2048
embedding size	1024
hidden size	1024
dimension of the feed-forward layer	4096
self-attention layers (for both encoder and decoder)	6
number of heads(multi-head self-attention mechanism)	16
dropout probabilities	0.3
merge operations(BPE)	32000
maximum number of tokens	4096
loss function	label smoothed cross entropy
adam betas	(0.9,0.997)
Maximum epoch number	50
warm-up steps	4000
Initial learning rate(Baseline system)	0.0007
Context-aware system(inside integration)	0.0001
Context-aware system(outside integration)	0.0001

4.2 Data Preprocessing

In Thailand-Chinese task, the data used in the experiment includes bilingual parallel corpus released by evaluation organization; some external data, such as bilingual sentence pairs and dictionary; monolingual data and pseudo parallel corpus. Bilingual parallel corpus is 200000 sentence pairs. 13069 sentence pairs and 1400 word pairs are collected from Internet as a supplement for bilingual parallel corpus. Monolingual data is extracted by similarity calculation between Chinese development set and CCMT'2021 Chinese monolingual database index by Elasticsearch[10]. Pseudo parallel corpus is generated by back translation system, whose source language sentence is from Chinese monolingual data.

Preprocessing method we adopted includes a general method and a specific method for given data. Both methods are used to reduce the data noise and improve the data quality[11]. The main stages of preprocessing are shown below.

- Traditional Chinese to simplified Chinese
- Full-width characters to half-width characters
- Special characters filtering
- Duplicating
- Sentence length filtering
- Sentence length ratio filtering
- Tokenization

Among above, in the process of sentence length filtering, we get Chinese sentence length by calculating the number of 'character' and get Thailand sentence length by the number of 'token' ,based on which we remove sentence pairs whose source sentence length or target sentence length exceeds the range of [1, 200]. Sentence length ratio filtering excludes the sentence pairs whose ratio of source sentence length and target sentence length exceeds the range of [0.1, 10]. In the tokenization stage, Thailand tokenization is implemented by Python tools Thainlp[12] and Chinese tokenization is implemented using the lexical tool Urheen[13].

Table 2. Preprocessing results of Training set data.

Type	Before preprocessing	After preprocessing
Bilingual parallel corpus	200000	191465
Dictionary	1400	1400
Bilingual sentence pairs	13069	6894
Pseudo parallel corpus	913432	901134
Chinese Monolingual data	1000000	913432

All steps of preprocessing are done on bilingual parallel corpus. Duplicating, sentence length filtering, and sentence length ratio filtering, Chinese tokenization are carried out on Chinese monolingual data. Sentence length filtering and

sentence length ratio filtering are implemented on the pseudo parallel corpus by back translation. Table 2 shows the data size comparison before and after preprocessing. 1000 sentence pairs are extracted respectively from the bilingual parallel corpus by evaluation organization as development set and test set. Finally 189465 sentence pairs are used as train set. All system below are trained on the development set and the test set. Their train set varies with different methods.

4.3 Experimental results

thc-2022-istic-primary-a model Baseline system and other context-aware systems are all trained on the given bilingual parallel corpus. Table 3 shows the results of baseline system and context-aware system under two methods(inside integration and outside integration) and two context (source context and target context). These models are all trained 50 epoch.Table 4 shows the effect of context-aware system is better than baseline system and the effect of the context-aware system under outside integration with target context is better than other system. So context-aware system under outside integration with target context is chosen as thc-2022-istic-primary-a model.This model's integrated target context in decoder is train set of Chinese and development set of Chinese.

Table 3. Performance comparison in different system

System	BLEU (test)
Baseline System	42.71
inside integration+source context	44.93
outside integration+source context	44.31
inside integration+target context	46.89
outside integration+target context	47.37

Table 4. Performance comparison in different training set

Mixing proportion(given corpus /pseudo corpus)	BLEU (test)
1:0	42.71
1:0.25	38.24
1:0.5	34.97
1:1	26.13
1:2	23.37
1:3	20.85
1:4	18.21

We adopted back translation method to generate pseudo parallel corpus. Context-aware system under outside integration with target context is trained on the released bilingual parallel corpus by evaluation organization, where source language is Chinese, target language is Thailand. 900000 Chinese sentences are filtered from monolingual data and translated into pseudo Thailand sentences. We mix the pseudo parallel corpus into other corpus in different proportions as new training set to train models.Context-based System (outside integration+target context) is trained on the above training sets. From the results in Table 4 pseudo corpus does not bring performance improvement of translation.

thc-2022-istic-primary-b model We adopt a model averaging strategy in the decoding phase and different results above are combined in post-processing stage to obtain the final translation. They make a model averaging and ensemble on thc-2022-istic-primary-c model and finally get a model whose Bleu scoring is the highest and choose it as thc-2022-istic-primary-b model.

Table 5. performance comparison of adding external corpus

Baseline training set	Dictionary	External sentences	BLEU (test)
189465	0	0	46.89
189465	1400	0	47.45
189465	1400	6894	47.62

thc-2022-istic-primary-c model We put Thai-to-Chinese dictionary and bilingual sentence pairs from Internet together with the released bilingual parallel corpus by evaluation organization as a new training set. Table 5 shows their performance comparison. From the results of Table 5, we can know external dictionary and bilingual sentence pair improve the translation effect. We choose this model as thc-2022-istic-primary-c model.

Table 6 shows the BLEU score[14] of three model submitted to evaluation organization.

Table 6. BLEU scoring test set (submitted models)

System	BLEU (test)
thc-2022-istic-primary-a model	47.37
thc-2022-istic-primary-b model	47.89
thc-2022-istic-primary-c model	47.62

4.4 Conclusion

This paper introduces the main and methods of ISTIC in CCMT' 2022. In summary, our model is constructed based on the Transformer architecture of the self-attention mechanism and a context-aware system. Although we tried the method of back-translation, it didn't work well. In terms of data preprocessing, several corpus filtering methods are explored. In the process of translation output, we adopt strategies such as model averaging and model ensemble. In the corpus filtering process, we use Elasticsearch to filter similar corpus. Experimental results show that these methods can effectively improve the translation quality. For machine translation tasks in low-resource languages, adding external dictionaries and parallel corpus can effectively improve translation performance. But in another view[15], it is worth exploring more to make more efficient use of small amounts of parallel training. Due to limited time, there are still many methods and techniques waiting us to exploit. Low-resource's neural machine translation is a very meaningful research problem. In the future, we will go into low-resource's neural machine translation and hope to make a contribution to it.

References

1. Ashish Vaswani et al.: Attention is all you need. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 30 (NIPS 2017), pp. 5998–6008. Neural Information Processing Systems, Online. (2017)
2. Jiajun Zhang, Chengqing Zong.: Neural Machine Translation: Challenges, Progress and Future. In: Science China Technological Sciences, vol 63, pp. 2028—2050. Springer, Heidelberg (2020). https://doi.org/10.1007/s11431-020-1632-x
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.
4. Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint
5. Elena Voita et al.: Context-aware neural machine translation learns anaphora resolution. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 1264—1274. Association for Computational Linguistics, Online. (2018)
6. 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.
7. Bei Li et al.: Does multi-encoder help? a case study on context-aware neural machine translation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3512—3518. Association for Computational Linguistics, Online. (2020)
8. Gerda Claeskens, Nils Lid Hjort.: Model Selection and Model Averaging. Cambridge University Press, Cambridge, United Kingdom. (2008). https://doi.org/10.1017/CBO9780511790485

9. Thibaud Lutellier et al.: CoCoNuT: combining context-aware neural translation models using ensemble for program repair. In: Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 101–114. Association for Computing Machinery, New York, United States. (2020)
10. Elasticsearch Homepage, https://github.com/elastic/elasticsearch. Last accessed 25 May 2022
11. Hangcheng Guo, Wenbin Liu, Yanqing He, Zhenfeng Wu, You Pan, Tian Lan, et al., ISTIC's Neural Machine Translation System for CCMT, 2021.
12. PyThaiNLP Homepage, https://github.com/PyThaiNLP/pythainlp. Last accessed 17 May 2022
13. Urheen, https://www.nlpr.ia.ac.cn/cip/software.html. Last accessed 15 May 2022
14. 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.
15. Rico Sennrich and Biao Zhang:Revisiting Low-Resource Neural Machine Translation: A Case Study. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 211–221, Florence, Italy. Association for Computational Linguistics.