Neural Machine Translation based on Back-Translation for Multilingual Translation Evaluation Task

Siyu Lai, Yueting Yang, Jin'an Xu†, Yufeng Chen and Hui Huang

School of Computer and Information Technology
Beijing Jiaotong University, Beijing, China
{20120374,20120442,jaxu,chenyf,18112023}@bjtu.edu.cn

Abstract. This paper presents the systems developed by Beijing Jiaotong University for the CCMT 2020 multilingual translation evaluation task. For this translation task, we need to build a Japanese-English translation system based on only Japanese-Chinese and English-Chinese data. Our method mainly relies on synthetic data generated by back translation. We implemented three different architectures, namely Transformer-big, Transformer-base and Dynamic-Conv. We also implemented multi-model ensemble technique to further boost the final result. Experiments show that our machine translation system achieved high accuracy without relying on any bilingual training data.

Keywords: Machine Translation, Multilingual Machine Translation

1 Introduction

This paper introduces in detail the submission of Beijing Jiaotong University to the multilingual translation evaluation task in the 16th China Conference on Machine Translation (CCMT2020). This task requires us to build a Japanese-English translation system based on only Japanese-Chinese and English-Chinese data. Due to the lack of direct training data, many techniques wildly used in the area of bilingual translation can not easily be applied in this scenario.

To train a translation model from Japanese to English, we created massive synthetic data based on two MT models of two different directions, namely Chinese-Japanese and Chinese-English [1]. Despite the lack of training data from Japanese to English, the training data for Chinese-Japanese and Chinese-English is readily accessible. The synthetic data can be obtained by translating the Chinese sentences to English of the Chinese-Japanese data, and the Chinese sentences to Japanese of the Chinese-English data. Further cleaning is applied to alleviate the noise contained in our synthetic data.

For the final Japanese-English model, we built our system based on three different architectures, the first one is solely based on attention mechanisms, namely the Transformer model [2]. We further augmented Transformer with deeper encoder layers, to better extract features from source segments, which is named as Transformer-big [3].

† Jinan Xu is the corresponding author.

Transformer-big was proved to outperform Transformer-base model in most cases. Additionally, we also tried to substitute the self-attention layer with dynamic convolution, providing us another different model to use when doing model ensemble [4].

Then, we applied sub-word segmentation to both languages to resolve the unknown words problem [5], as well as model ensemble, to leverage multi-models to further improve the result [6]. Moreover, we did some contrast experiments, and the results show that our machine translation systems achieved high accuracy without relying on any bilingual data, performed better than other models, proving the effectiveness of our training procedure.

2 Related work

Recent several years, neural machine translation (NMT) [7-8] performs end-to-end translation based on an encoder-decoder framework and works well in many machine translation tasks. In this framework, the encoder firstly transforms the source sentence into a fix-length vector, and the decoder generates a target sentence. Such framework has achieved significant improvements over traditional SMT with abundant parallel data available.

However, high-quality and large-scale parallel corpora are non-existent for most circumstances. Lots of work have been done to address this problem, which can be divided into two broad categories: *multilingual* and *pivot-based* approaches.

Johnson [9] has proposed a universal NMT model to translate between multiple languages without any changes in the model architecture, which took full use of multilingual data to improve NMT for all relevant languages. Firat [1] has proposed a multilingual model consisting of several encoders and decoders with finetuning algorithm. It was really difficult to learn and analyze the universal representation for multiple languages, although they have completed direct source-to-target translation without using parallel corpora.

Another crucial way is to bring in a third language named *pivot*, acting as a bridge between the source and the target language. Although it is hard to find available in-domain parallel data, the parallel corpora with a pivot language usually exist. For instance, the parallel data between Japanese and English directly is rare, but the parallel data between Chinese and Japanese, Chinese and English is relatively rich. In pivot-based machine translation, sentences are translated from the source language into pivot language firstly, then translated from the pivot language into the target language.

Although pivot-based method performs well in most translation tasks, it still has some disadvantages. Firstly, the mistakes made in source-to-pivot translation will be propagated to pivot-to-target step which can cause error propagation problem. Furthermore, translated by this pipeline way may lose some relevant information in the pivot translation and may not be represented in the target sentence.

The inspiration for our work came from Sennrich [10], and we introduced a pivot language, via which we could make full use of large bilingual corpora. By back translation, we could create synthetic data for training final model.

3 Model

As we explained above, we combined three different architectures in our work — Transformer-base, Transformer-big and Dynamic-Conv.

3.1 Transformer-base

The first model we used is Transformer-base, which is a completely attention-based structure for dealing with problems related to sequence models, such as machine translation. The Transformer model dispenses with any CNN or RNN structure, capable of working in the process of highly parallelization, so the training speed is very fast while improving the translation performance.

The structure of Transformer is shown in Figure 1. The model is divided into two parts: encoder and decoder. The encoder consists of six identical layers, each with two sublayers. The first sub-layer is the self-attention layer, and the second sub-layer is a simple fully connected feedforward network.

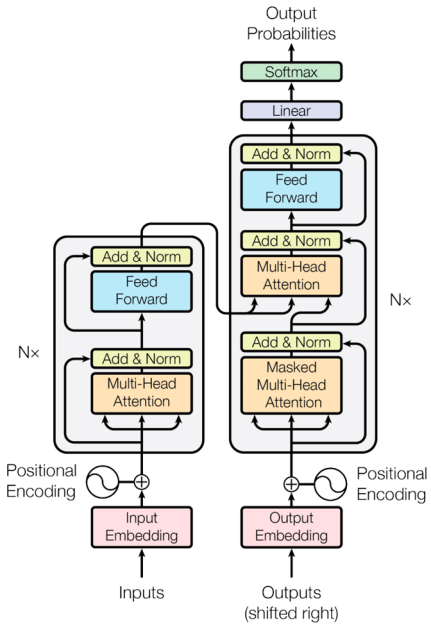

Fig. 1. The Transformer - model architecture.

Residual connections are added outside the two layers, and then layer normalization is performed. In addition to the two layers in the encoder, the decoder also adds a third sub-layer to connect the encoder and decoder. As shown in the figure, the decoder also uses residual error and layer normalization. The output of each sub-layer is:

$$output = \text{LayerNorm}(x + (\text{SubLayer}(x)) \quad (1)$$

The particular attention is called *Scaled Dot-Product Attention*, which takes queries keys of dimension d_k and values of dimension d_v as input, calculates the attention function on a set of queries simultaneously, and packs them together into a matrix Q. The keys and values are also packed together into matrices K and V. The output of the matrix can be calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

Multi-head attention allows each head to acquire separate attention weights from different representation subspaces at different position.

$$\begin{aligned}\text{MultiHead}(Q, K, V) &= \text{Concat}(head_1, \dots, head_h)W^O \\ \text{where } head_i &= \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)\end{aligned} \tag{3}$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

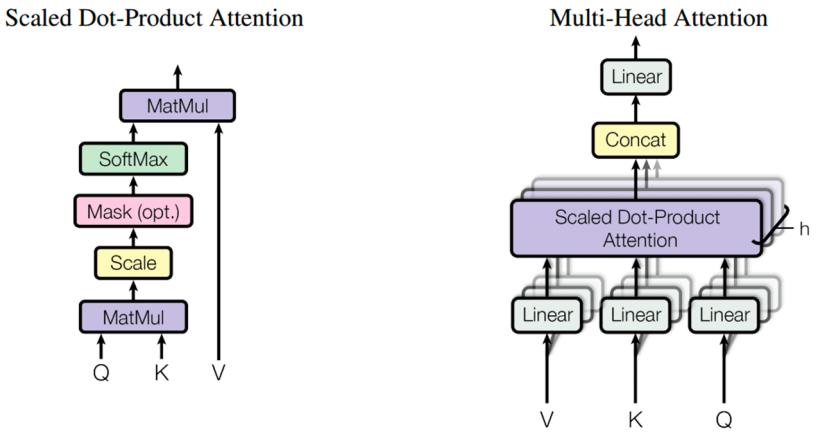

Fig. 2. (left) Scaled Dot-Product Attention. (right) Multi-Head Attention.

Since transformer model does not use any CNN or RNN structure, they introduce some information with relative or absolute position of tokens in the sequence, in order to take advantage of the order information. The position encoding is defined as:

$$\begin{aligned}PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{model}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{model}}\right)\end{aligned} \tag{4}$$

Where *pos* is the position and *i* is the dimension.

3.2 Transformer-big

To boost its ability to extract features and provide a better presentation for source segment, we deepen the encoder layers for Transformer, and this model is called Transformer-big. Our Transformer-big contains 12 layers of encoder, while Transformer-base only contains 6 layers. However, more encoder layers may encounter the vanishing-gradient problem and entail extra strategies.

3.3 Dynamic-Conv

Self-attention is an effective mechanism. Since it was proposed, it has been applied to many NLP tasks with good performance improvement. However, for long sequences, self-attention is limited by its $O(n^2)$ complexity. In addition, the feature that self-attention can efficiently capture long-term dependence has also been questioned. Therefore, a new structure called *lightweight convolution* is proposed to replace self-attention with CNN structure.

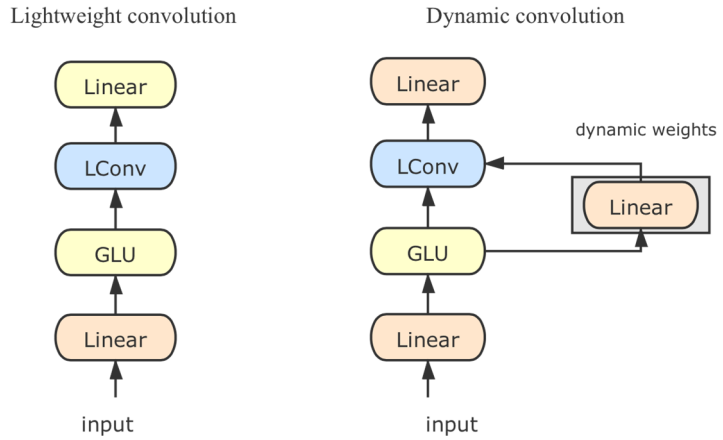

Fig. 3. (left) Lightweight convolution. (right) Dynamic convolution.

Lightweight convolution uses the prototype of deep (separable) convolution in CV domain, which greatly reduces the number of parameters and complexity by sharing parameters in the channel dimension. Based on the Lightweight, dynamic convolution is proposed, where the weight of CNN is calculated dynamically from the input feature, as shown in Figure 3. The Dynamic-Conv model is proved to be competitive with Transformer model in many scenarios.

$$\text{DynamicConv}(X, i, c) = \text{LightConv}\big(X, f(X_i)_{h,:}, i, c\big) \tag{5}$$

Where f is a simple linear module with learned weights $W^Q \in \mathbb{R}^{H\times k\times d}$, i.e., $f(X_i) = \sum_{c=1}^{d} W^Q_{h,j,c} X_{i,c}$.

4 Experiments

4.1 Preprocessing

Since the quality of training data is vital for the final system, we cleaned the provided training data according to the following strategies:

1. Remove sentences containing too many garbage characters;
2. Remove sentences too long or too short;
3. Remove sentence-pairs with a length ratio too big or too small;
4. Remove duplicate sentence-pairs;
5. Remove sentence-pairs with a too low alignment score provided by fast-align[1];

Both Chinese-Japanese and Chinese-English parallel corpora provided by the organizer contained 3 million sentence pairs. After doing the 5 preprocessing steps mentioned above, 2.99 million sentence pairs were left in each dataset.

And then we used Jieba[2] to perform Chinese word segmentation, and nltk[3] to tokenize English, and Mecab[4] to do Japanese word segmentation. To alleviate the out-of-vocabulary problem and reduce the vocabulary size, we applied sub-word segmentation for both languages, provided by subword-nmt[5].

4.2 Back Translation based Synthetic Data

To train a translation model from Japanese to English, the parallel corpus from Japanese to English is in need. However, only the parallel data of Chinese-English and Chinese-Japanese are provided. To create synthetic data for the training of final Japanese-English model, we used back translation.

Back translation does not need to make any change in the training algorithm and the model network structure. Back translation has been proved to be simple but effective, while sometimes we may get particularly outrageous translation results in the process of back-translation. Our whole training procedure contains the following steps:

1. Train a Chinese-English model and a Chinese-Japanese model using the parallel data provided.
2. Translate the Chinese sentences in Chinese-Japanese data into English using Chinese-English MT model.
3. Translate the Chinese sentences in Chinese-English data into Japanese using Chinese-Japanese MT model.

---

1 https://github.com/clab/fast_align
2 https://github.com/fxsjy/jieba
3 http://www.nltk.org/
4 https://github.com/SamuraiT/mecab-python3
5 https://github.com/rsennrich/subword-nmt

4. Combine the synthetic Japanese-English data of step 2 and step 3 together, and train the final Japanese-English model.

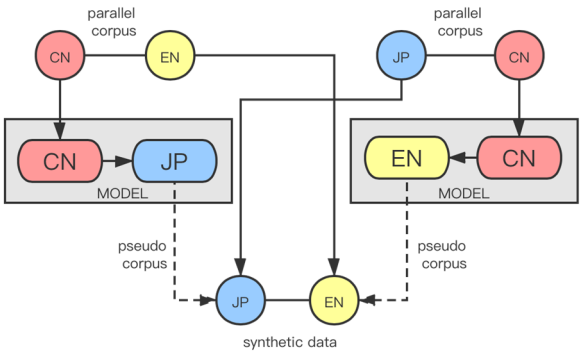

Fig. 4. Back-Translation procedure

The training steps above were implemented on all of three architectures. Since the synthetic data was generated by our own machine translation model, which means the translated side contained a large amount of noise, we performed the following cleaning steps:

1. Remove sentence-pairs with low language model scores on the target side provided by kenlm[6];
2. Remove sentence-pairs with low alignment scores provided by fast-align;

In the first step, we kept sentences with kenlm scores from -10.0 to -200.0. In the second step, we kept sentences with fast-align scores greater than -500.0. After combining two synthetic datasets, we finally got 5.81 million sentence pairs as training dataset of Japanese-English model.

There are also other ways to deal with the absence of bilingual data, such as pipelined-training and hybrid-labels [9]. Previous evaluation participants and the contrast experiments we did demonstrated that the back-translation based method is normally the most effective while easy to implement.

4.3 Multi-model Ensemble

For multi-model ensemble, we tried the strategy of Independent Parameter Ensemble (IPE), that is to firstly train several models with different architecture and different initialized parameters, and then weight the average probability distribution of multiple models in the Softmax layer. Better models are assigned with relatively higher weights, and worse models with relatively lower weights.

---

[6] https://github.com/kpu/kenlm

4.4 Contrast Experiments

In order to prove the superiority of the back-translation based model more comprehensively, we did some contrast experiments like pipeline method, sequence-level knowledge distillation [11] and domain adaptation [12]. Pipeline method firstly translated Japanese to Chinese, and then from Chinese to English. Knowledge distillation used right-to-left and target-to-source model to decode training data, then combining it with synthetic Japanese-English data we generated previously, and used this new data to train model from scratch. Domain adaptation used English sentences in synthetic Japanese-English data to train in-domain model and used English monolingual corpus to train general-domain model, calculating the absolute value of the subtraction between two language models and remaining corpus with low difference value.

4.5 Results

Experiment results on the development set are shown in Table 1. (evaluated by sacreBLEU)

Table 1. Experiments on Development Set

Method	Model	BLEU
Pipeline	Transformer-base	31.29
Knowledge Distillation	Transformer-base	34.47
Domain Adaptation	Transformer-base	34.90
Back-Translation	Transformer-big	35.11
	Transformer-base	35.24
	Dynamic Conv	34.9
	Ensembled	**35.66**

Official automatic evaluation results are shown in Table 2.

Table 2. Official automatic evaluation results

	BLEU5-SBP	BLEU5	BLEU6
je-2020-bjtu_nlp-primary-a	38.29	40.47	35.81

4.6 Model Analysis and Discussion

As shown in table 1, it's obvious that back-translation based model did better than other methods. The reason is that the pipeline method will cause error propagation and lose some relevant information, but back-translation based method does not. As for knowledge distillation, because our task is based on patent domain, it is possible that the teacher model is not strong enough to guide the student model. For domain

adaptation, remaining corpus with low difference value cannot guarantee the quality of the data. It is possible that both of two language model have low scores and the difference value is relatively small, so the remaining corpus may affect the quality of data. Moreover, using selected monolingual corpus to generate pseudo corpus may damage the quality of data again. Therefore, we can conclude that back-translation is a simple and effective approach to multilingual translation task.

5 Conclusion and Future Work

In this paper, we described our submission in multilingual translation evaluation task. For this translation task, we need to build a Japanese-English translation system based on only Japanese-Chinese and English-Chinese data. Our method mainly relies on synthetic data generated by back translation. We implemented three different architectures, namely Transformer-big, Transformer-base and Dynamic-Conv. We also implemented multi-model ensemble technique to further boost the final result. Experiments show that our machine translation system achieved high accuracy without relying on any bilingual training data.

We have to mention that we also tried other strategies which are commonly used in bilingual translations, including domain adaptation, sequence-level knowledge distillation and checkpoint ensemble [13], but none of them made it to introduce any improvement. Even multi-model ensemble could only introduce minor improvements. This may be caused by the pivot-based synthetic data, and we will explore this problem in our future work.

Acknowledge

This work is supported by the National Natural Science Foundation of China (Contract 61976015, 61976016, 61876198 and 61370130), and the Beijing Municipal Natural Science Foundation (Contract 4172047), and the International Science and Technology Cooperation Program of the Ministry of Science and Technology (K11F100010).

References

1. Firat, O., Sankaran, B., Al-Onaizan, Y., Vural, F. T. Y., Cho, K.: Zero-resource translation with multi-lingual neural machine translation. arXiv preprint arXiv:1606.04164 (2016).
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I.:Attention is all you need. In Advances in Neural Information Processing Systems pp. 5998-6008 (2017)
3. Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D. F., Chao, L. S.: Learning deep transformer models for machine translation. arXiv preprint arXiv:1906.01787 (2019)
4. Wu, F., Fan, A., Baevski, A., Dauphin, Y. N., & Auli, M.: Pay less attention with lightweight and dynamic convolutions. arXiv preprint arXiv:1901.10430 (2019)

5. Sennrich, R., Haddow, B., Birch, A.:Neural machine translation of rare words with subword units. In Proceedings of ACL (2016)
6. Rokach, L.: Ensemble-based classifiers. Artificial intelligence review, 33(1-2), 1-39 (2010).
7. Kalchbrenner N, Blunsom P. Recurrent continuous translation models[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. In Proceedings of EMNLP,2013:1700-1709.
8. Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014:1-15.
9. Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z.: Google's multilingual neural machine translation system: enabling zero-shot translation. Transactions of the Association for Computational Linguistics, 5, 339-351 (2017).
10. Sennrich, R., Haddow, B., Birch, A.:Improving Neural Machine Translation Models with Monolingual Data. In Proceedings of ACL. (2016)
11. Kim, Y., Rush, A., M.:Sequence-Level Knowledge Distillation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas. In Association for Computational Linguistics. (pp. 1317-1327) (2016)
12. Axelrod, Amittai & He, Xiaodong & Gao, Jianfeng. Domain Adaptation via Pseudo In-Domain Data Selection. EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. 355-362. (2011)
13. Chen, H., Lundberg, S., & Lee, S. I.: Checkpoint ensembles: Ensemble methods from a single training process. arXiv preprint arXiv:1710.03282 (2017).