The En-Ru Two-way Integrated Machine Translation System Based on Transformer

Anonymous ACL submission

Abstract

Machine translation is one of the most popular areas in natural language processing. WMT is a conference to assess the level of machine translation capabilities of organizations around the world, which is the evaluation activity we participated in. In this review we participated in a two-way translation track from Russian to English and English to Russian. We used official training data, 38 million parallel corpora, and 10 million monolingual corpora. The overall framework we use is the Transformer(Vaswani et al., 2017) neural machine translation model, supplemented by data filtering, post-processing, reordering and other related processing methods. The BLEU(Papineni et al., 2002) value of our final translation result from Russian to English is 38.7, ranking 5th, while from English to Russian is 27.8, ranking 10th.

1 Introduction

Neural machine translation has been widely used in the field of machine translation, because it is more accurate than statistical machine translation in most cases. The proposed attention mechanism brought a new revolution in the neural machine translation, making the overall effect of translation much better than before. Then, the Transformer that makes full use of the attention mechanism , both in terms of performance and effectiveness. Up to now, most of the work has been carried out on Transformer, and its superiority has been widely recognized.

From the beginning of machine translation research, there has been the development of two-way translation between Russian and English. As early as 1954, Georgetown University in the United States under the IBM company completed the English-Russian machine translation experiment with IBM-701 computer, which opened the prelude of machine translation research. During the period, there are three core technologies, rule-based machine translation, statistical machine translation(Koehn et al., 2007) and neural machine translation(Bahdanau et al., 2014), which continue to develop. However, as the application fields of machine translation become more and more complex, the limitations of different technologies begin to appear. Because of the more application scenarios and the higher requirements for accuracy, the problem of model optimization appeared.

The translation between Russian and English is extremely difficult because their linguistic features are distinguished and the lexical composition and grammatical structure of Russian are more complicated than English. Early statistical machine translations were hoped to be implemented through phrase-based methods(Marcu and Wong, 2002), including rule-based lexical, phrase analysis systems, and related techniques for language models and translation models. These methods have solved the translation problem between Russian and English to a certain extent. However, at the same time, there is still a problem that the time cost is long and the translation result is not good enough.

Therefore, the emergence of neural machine translation has brought a new dawn for the translation between Russian and English. The basic modeling framework for neural machine translation is an end-to-end sequence generation model, a framework and method for transforming input sequences into output sequences. There are two points in the core part. One is to represent the input sequence through the encoder, and the other is to obtain the output sequence through the decoder. In addition, for machine translation, neural machine translation not only includes encoding and decoding, but also uses RNN(Sutskever et al., 2014) or other methods to encode sentence

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 434–439
Florence, Italy, August 1-2, 2019.

pairs. It also introduces an additional mechanism, the attention mechanism(Luong et al., 2015), to help us to convert sequences. The translation results thus obtained more expectations than before. Later, Transformer appeared, which greatly enhances neural machine translation in terms of performance and effect.

This paper is based on Transformer, a neural machine translation network structure, to develop a two-way evaluation task between Russian and English. Taking into account the language characteristics of Russian and English, we have done appropriate operations in data preprocessing, including removing duplicates, deleting unreasonable sentence pairs, lowercase and Latinization operations, and judging sentence alignment problems, removing the parallel corpus with problems. The filtered parallel corpus is then sent to the model for training and the training results are tested. After getting the trained model, we start to consider using the back-translation operation to augment the data, continuing to filter the generated artificial corpus, and put it into the model training together with the original parallel corpus.

Finally, ensemble(Dietterich, 2000), average and rerank(Shen et al., 2004) operations are implemented on different models to improve the overall performance of the translation system.

2 Background

Neural network machine translation is based on a sequence-to-sequence overall structure consisting of an encoder and a decoder. The encoder converts the source language sentence into an intermediate sequence result, and the decoder converts the intermediate sequence result into a target language sentence. There is also the Attention mechanism to help make the results perform better. In the construction of the overall translation system, we used a lot of excellent methods proposed by the predecessors.

The basic model used here is Transformer. This is a paper published by Google in 2017 titled Attention Is All You Need, an attention-based structure proposed to deal with sequence model related issues, such as machine translation. Traditional neural machine translation mostly uses RNN or CNN as the model base of encoder-decoder, and Google's latest Attention-based Transformer model abandons the inherent formula and does not use any CNN or RNN structure. The model works in high-level parallel process, so training speed is also extremely fast while improving translation performance.

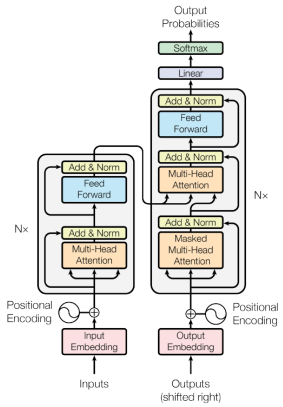

Figure 1: Transformer Structure

The structure of Transformer is shown in Figure 1. The model is divided into two parts: the encoder and the decoder. The encoder is stacked by six identical layers, each with two more sub-layers. The first sub-layer is a long self-attention mechanism, and the second sub-layer is a simple fully connected feed forward network. A residual connection is added outside the two layers, and then layer normalization is performed. The output dimensions of all sub-layers and embedding layers of the model are d_{models}; the decoder also stacks six identical layers. However, in addition to the two layers in the encoder, the decoder also adds a third sub-layer, as shown in the figure which also uses the residual and layer normalization.

3 Experiment

For this evaluation task, we start from the data preprocessing, through the data augmentation operation, get the parallel corpus that needs to be trained, input the Transformer model for training, and test the training results, and finally ensemble results according to the model generated by different strategies, average and rerank operations, for the best results. Next, the experimental content will be elaborated separately. The overall experimental process is shown in Figure 2.

3.1 Data Preprocessing

The first is data preprocessing, which is crucial for the translation of the model. The sentences used in this evaluation with data preprocessing method to filter out include parallel sentence pairs with

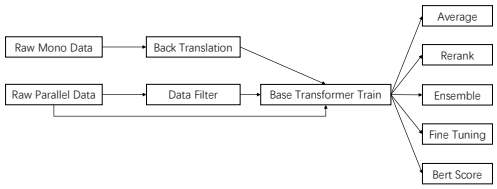

Figure 2: Project Process

high repetition rate, length mismatch and alignment problems. The amount of data given by the official at the beginning was about 38 million lines. After data filtering, 33 million lines were left, and 5 million lines were deleted, accounting for 0.13 of the original quantity. This result is in line with expectations and acceptable.

The sentence with higher repetition rate has little meaning in the training corpus, which increases the burden of the model and affects the translation effect, so it needs to perform deduplication operation. The method used here is to calculate the cosine distance of the SimHash value between each row of data. When the difference is less than 0.2, we believe it is repeated, and can be deleted. Because the amount of data is large and the global deduplication time is too long, so here is a simple calculation of three sentences before and after current line, that is, using a window of size 7 to check the sentence repetition, which also conforms to the principle of local consistency.

From the practical experience and linguistic knowledge, the length of sentences generated by the two languages expressing the same meaning is not too different, especially for Russian and English. So we also screened the length of the sentence. In the experimental processing, we control the ratio of the source language and the target language length to 1/2-2/1, which means that the sentence lengths of the two languages are not more than twice as large. The length of the sentence is calculated by the number of tokens. The parallel sentence pairs thus obtained are also reasonable in length ratio.

Sentence alignment is a very important factor to measure the quality of parallel sentence pairs from the perspective of sentence meaning. Different from the previous method, it needs to enter sentences themselves and judge whether the data pairs are reasonable according to the correspondence between words in the two languages. The gize++ tool(Gao and Vogel, 2008) is used here to help check for data alignment issues. By reading Russian-English vocabulary and Russian-English parallel corpus information, creating a new dictionary, building an IBM model 1, making EM algorithm iteration, generating word alignment information, and obtaining a calculated sentence pair for each data. We generate alignment scores and eliminate sentence pairs with scores less than the threshold 10e-10 for better alignment data.

3.2 Back Translation

In the process of data augmentation, the back-translation strategy(Edunov et al., 2018) plays a crucial role. The auxiliary translation system from the target language to the source language first trains on the available parallel data and then uses to generate translations from the monolingual corpus of the large target. The pairs of these translations and their corresponding reference targets are then used as additional training data for the original translation system. Using this strategy can greatly increase the data required for training and improve the translation effect of the model. In the back-translation, we trained a translation model from the target language to the source language based on the existing corpus. By inputting the target language corpus into the model, the corresponding source language corpus can be obtained, and the two are combined to obtain a new parallel corpus.

The data set size of this trial is not too large and it is stipulated that external parallel corpus expansion cannot be used, so we use the back translation method to increase the amount of training data.

Using back translation extended corpus in NMT is a common data enhancement technique. We trained a translation model from the target language to the source language based on the existing corpus. By inputting the target language corpus into the model, the corresponding source language corpus can be obtained and combined to get a new parallel corpus.

External data is not allowed in this competition, so we use the mono part of the original corpus to generate para data. However, there is a problem with this approach that there may be duplication between the new parallel corpus and the original corpus. To solve this problem, we added some

random noise on the decoding side to avoid this situation.

We selected 10 million Russian and English sentences respectively from the official monolingual corpus as raw data for back translation operations. The model obtained through the training of the existing parallel corpus translated this part of the monolingual corpus and obtained 10 million pseudo-parallel corpora. Then, we filter this part of the data according to the data filtering and noise strategy mentioned in the previous section. Finally, 8 million individual parallel corpora are obtained and the filtered parallel corpus input model is used for training operations.

3.3 Model training

Considering the hardware cost and time cost of the experiment, the model we selected for this experiment is the basic version of Transformer. The encoder and decoder have 6 sub-layers and the multi-head attention mechanism has 8 headers. The word vector size is 512. Guaranteed to get the best results in a limited time in a laboratory environment. The development environment for evaluation is MXNET, which is the deep learning library that Amazon chose.

The input model needs to be further processed before training, including generating the corresponding token for the sentence. The tool used here is the commonly used tokenizer.perl, which can separate the words and punctuation in English and convert the special symbols to keep the same symbol. Russian is the same. In addition, the BPE method is needed to generate the subword vocabulary to reduce the vocabulary size during the model training and improve the performance of the model.

After the above processing, the data can be divided to obtain a training set, a test set and a verification set, the training set is used for model training, the verification set is used for performance detection in the training process, and the test set is used for evaluating the result of model trained.

For the evaluation task, the following experiment was designed:

1. Baseline Model

Use the official 38 million parallel corpus without screening and direct it into the model for training and testing. The results of the base model are used to compare with different strategy results and generate reverse translation data to extend the corpus and continue training. The purpose is to maintain the generalization ability and robustness of the model to the greatest extent, and to provide reference for other model training results.

2. Filter Model

The data preprocessing operation is used to screen the official data and the ideal training corpus is obtained. The 33 million filtered parallel sentences are trained to obtain a data filtering model. Because the quality of the data used for training is higher, the effect of model translation is better than the basic model.

3. Back Model

10 million is extracted from the official monolingual corpus as the source language input to the baseline model for translation, and the artificial parallel corpus based on the baseline model translation is obtained. Since the effect of the baseline model is not good enough, the generated corpus needs to be further filtered, and the method is also the data preprocessing operation mentioned above. After screening, we got about 8 million good quality artificial data and then combined the artificial parallel corpus with the previously filtered official parallel corpus and input them into the model for training. Then we got the Back translation model. Because artificial corpus has been added, the translation effect and the robustness is improved.

4. Fine-tuning Model

Fine-tuning a trained model using small-scale corpus is a commonly used strategy in the field of machine learning. It can make the model more sensitive to specific domain scenarios, thus reflecting better results. Here, we select a corpus with much similarity to the test set from the training set to fine-tune the trained model. The similarity scores between the test corpus and the training corpus are sorted and ranked. Then the parallel sentence pairs with higher scores are found and the corpus is extracted as a fine-tuning corpus. In this way, about 5,000 pieces of data are obtained and this part of the corpus is input into the previously trained model to obtain the result of fine-tuning the model, so that it can perform better on the test set.

5. Ensemble Model

Ensemble is a method that combines the results of multiple models. The purpose of this is to complement the advantages of different models, make up for the problems that fall into the local optimum and get the results of the machine translation

model with better comprehensive effects. For the sake of simplicity, only different initialization random seed parameters are set for the same model. So training of multiple models is performed, generally two or three models, and finally the results of all models are subjected to ensemble operation. By composing and complementing multiple models, we obtain the comprehensive optimal results of data translation.

6. Average Model

The Average operation is similar in thought to ensemble, but it operates on different training parameters of the same model. The parameters in these training results are subjected to the average operation, and a set of training results are comprehensively obtained from the best training parameters of the single model. Top 5 of the model training results is selected for averaging to prevent a certain result from falling into the local optimum and a plurality of parameters are integrated to obtain an averaged optimal solution. This results in the best combination of different training parameters in the same model, thereby improving the performance of a single model.

7. Nbest and Rerank Model

Extracting only one of the highest-scoring statements from the translation results of the model as an output is not necessarily the best result. So this strategy can be used to extract the best three from each translation model result as a candidate set. Then use some rules to rerank and get the best one as the output result. The translated content thus obtained is the comprehensive output of multiple results of each model, which is theoretically optimal. The rules used here include weighted summation of beam search score and the language model scores. The first one is based on the beam score returned during decoding, but different models have different performances, so it is difficult to sort under a uniform metric. So we introduced different weights for different models. Using beam score weight as the final score for each translation result, the final result was obtained by screening. The second one gives scores of the generated translations using the pre-trained language model. They are judged from the linguistics itself and the sentences with the highest scores are selected. The final result is an output that combines the highest scores of the two methods described above.

The above models also had different batch sizes, comparison of the number of graphics cards and vocabulary sizes in the training process. We extracted them for the optimal results. Finally, the output is simply post-processed. In order to comply with normal text habits. However, due to the limitations of time and hardware resources, not every experiment has been refined and detailed totally, so there is still improvement of results in the future.

Name	Pair	Bleu	Improve
base-re	RU-EN	34.8	0
filter-re	RU-EN	36.1	+1.3
average-filter-re	RU-EN	36.2	+1.4
rerank-re	**RU-EN**	**37.5**	**+2.7**
vote-re	RU-EN	36.1	+1.3
base-er	EN-RU	25.6	0
filter-er	EN-RU	26.6	+1.0
average-filter-er	EN-RU	26.8	+1.2
rerank-er	**EN-RU**	**27.8**	**+2.2**
vote-er	EN-RU	26.5	+0.9

Table 1: Experiment Result.

3.4 Results Analysis

The above experimental results are presented in the Table 1. It should be noted that only the better and more complete results in the experiment are given here. We can see that the BLEU values of the Baseline Model form English to Russian is 25.9, while from Russian to English is 35.2, respectively as a benchmark, to provide reference for the following models. The results after filtration are 27.0 and 36.5, which has 1.0 or so improvement over baseline. The results obtained by the Average strategy are 27.2 and 36.5, which is basically no improvement. The strategy of obtaining nbest for translation results and reranking according to the reference rules worked very well, which got 28.2 and 38.0, from 2 to 3 points higher than baseline. Back Model, Fine-tuning and Ensemble strategies are not very completed in detail, so they are not shown here.

4 Conclusion

In this evaluation task, we established a Russian-English two-way machine translation system based on Transformer. Through data preprocessing, model training, data post-processing and other optimization strategies, the evaluation results were finally from English to Russian BLEU value 28.2, while from Russian to English 38.0, which was

about 3 points higher than the baseline result. In the final list, we got 5th in Ru-En, and 10th in En-Ru. Good results have been obtained in limited time and hardware resources, which is also in line with the industry's demands for service construction. In the whole experiment process, we also learned a lot of experience in data processing and experimental design, which will be of great help in later research and study. We will continue to improve the previous experiments, strive to get better results, and see what rankings can eventually be achieved, in preparation for the next year.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. *Software engineering, testing, and quality assurance for natural language processing*, pages 49–57.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Daniel Marcu and Daniel Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.