Neural machine translation model combining dependency syntax and LSTM

Xin Zheng, *Hailong* Chen*, *Yuqun* Ma, and *Qing* Wang

School of Computer Science and Technology, Harbin University of Science and Technology, Heilongjiang, China

Abstract. For the problem of the lack of linguistic knowledge in the neural machine translation model, which is called Transformer, and the insufficient flexibility of positional encoding, this paper introduces the dependency syntax analysis and the long short-term memory network. The source language syntactic structure information is constructed in the neural machine translation system, and the more accurate position information is obtained by using the memory characteristics of LSTM. Experiments show that using the improved model improves by 1.23 BLEU points in the translation task of the IWSLT14 Chinese-English language pair.

Keywords: Neural machine translation, Transformer, Long and short-term memory network, Dependency syntax.

1 Introduction

Machine translation (MT) refers to how a computer translates a source language sentence into a target language sentence with the same semantics[1] and is an essential branch in natural language processing. People divide Machine translation into three types: rule-based machine translation, statistical machine translation (SMT), neural machine translation (NMT)[2]. With the continuous development of deep learning, neural machine translation has outperformed statistical machine translation in most language pairs and has become the mainstream in machine translation[3]. In 2017, Vaswani et al.[5] proposed a Transformer model based on the attention mechanism, although it significantly improved the training speed and translation quality. However, there are still some shortcomings:

A neural network-based translation model such as a recurrent neural network is a sequential structure that contains the position information of words in the sequence. While Transformer does not contain recurrent neural network (RNN) and convolutional neural network (CNN), the transformer can not have the ability to identify the order of each token, and does not contain any position information. Adding Positional Embedding (PE) can help the model identify positional information. The Sinusoidal positional embedding proposed by Vaswani et al. uses a predefined function to calculate the positional embedding information, which does not contain learnable parameters and is not flexible enough.

* Corresponding author: hrbustchl@163.com

The neural machine translation model adopts the encoder-decoder framework, an end-to-end model[7]. In neural machine translation, neural machine translation models treat source language sentences as words or sequences of words, ignoring the structural information inherent in the language. The first explicit introduction of syntactic knowledge in neural machine translation started with Eriguchi et al.[8], which showed that introducing linguistic knowledge into neural machine translation would significantly improve translation performance.

For the above problems, this paper uses Transformer as the benchmark model and uses the dependency syntax tree [9] to obtain the minimum distance between each word, which is the closeness of the relationship between words and words, and then converts it into the corresponding dependency matrix. Finally, based on the CBOW[10] model, predicting the target word according to the dependent word is added to obtain the word vector containing the dependency. Using the LSTM model [11] trained the output vector. At this time, the input sequence contains each sentence's semantic structure and word position information. In this paper, it is verified through experiments that the above methods can improve translation performance. Finally, the evaluation results of the improved model in this paper on the test2010-2013 test set verify the method's effectiveness.

2 Related work

Due to some shortcomings of the original transformer model, many researchers in the academic community began to improve the Transformer. Some related works have been tried from different angles, using different methods to construct position encoding information, roughly divided into absolute position encoding and relative position encoding, and some other fancy encoding methods.

Since natural language is generally more dependent on relative position, Shaw et al. [6] proposed relative position encoding, which does not fully model the position information of each input but is an improved extension of the self-attention mechanism to consider the method for pairwise relationships between elements. The relative position representation is used in the Transformer's self-attention mechanism. When training Attention, the relative distance between the current position and the position of the Attention is considered, which dramatically improves the translation performance of the neural machine translation system. Although relative position encoding has no limit on text length, it sacrifices long-distance position dependence to some extent. Wang et al. [13] adopted a dependency tree to represent the semantic structure of a sentence, which simplified the syntactic relationship between input words and encoded positional information according to the depth of each word in the dependency tree. Chen et al.[14] proposed a word-vector-based recursive positional embedding method to capture sequential dependencies based on word content in sentences. Employs recurrent positional embeddings learned by a recurrent neural network to encode word content-based order dependencies into word embeddings. They are then integrated into existing multi-head self-attention models as independent heads or as part of each head.

In integrating syntactic analysis into neural machine translation, Li et al. [16] performed depth-first traversal of the syntactic tree of source language sentences and linearized it to obtain syntactic structure information. Sennrich et al. [17] used the source language dependency syntactic structure to use the dependency relationship, part of speech, a word root, and other information in the dependency structure as features, which were represented by different vectors and spliced together with the word vector to form each source language word. the input vector.

Inspired by the advantages and disadvantages of the above-mentioned related work, this paper proposes DTCL (dependency tree combined LSTM). Integrate the dependencies into the input sequence, and use the memory characteristics of LSTM to concatenate the output

of each time step, that is, the position information of each word and the original sequence, as the input sequence of the entire Transformer. This method solves the shortcomings of the original model.

3 Background

The transformer is a classic NLP model proposed by Google's team in 2017. It only uses the Attention mechanism for machine translation tasks and has achieved good translation results. The transformer uses the encoder-decoder.

As said above, the Transformer processes each word in parallel and cannot identify the order of each token. Adding positional coding can help the model identify positional information. Therefore, to take advantage of the order of the sequence, the positional encoding is added to the input sequence at the bottom of the encoder and decoder stacks. The positional encoding has the same dimension as the word vector, and the two can be directly added as the encoder input. Use sine and cosine functions of different frequencies, as shown in equations (1) and (2).

$$\mathrm{PE}_{(POS,2i)} = \sin(pos / 10000^{2i/d_{\mathrm{model}}}) \tag{1}$$

$$\mathrm{PE}_{(POS,2i+1)} = \cos(pos / 10000^{2i/d_{\mathrm{model}}}) \tag{2}$$

Where pos is the element position in the sequence, $2i$ is the current dimension of the position encoding vector, and d_{model} is the dimension of the input sequence. Each dimension of the positional encoding corresponds to a sinusoid. The wavelengths form a geometric progression from 2π to $10000 \cdot 2\pi$.

According to equations (1) and (2), it can be seen that the Transformer model has limitations in obtaining position encoding information. According to the memory characteristics of LSTM, this paper captures long-distance dependence to obtain position information, and each time series output contains the information of the previous word to obtain the position code.

The Transformer model architecture is shown in Figure 1.

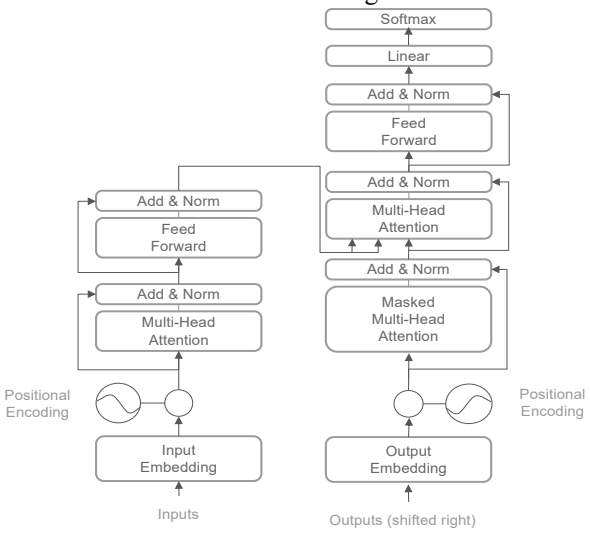

Fig. 1. Transformer model architecture.

4 Neural machine translation system integrating dependency syntax and LSTM

This section describes methods for incorporating dependency syntax and LSTMs into Transformers. This paper only makes the same improvements to the input and output sequences of the benchmark model Transformer. The source language sentence word vector is passed through the dependency syntax tree to make the obtained input sequence contain semantic information and then input to the long and short-term memory network model to obtain the position information code. Finally, the word vector and the position information code is spliced into the final input sequence.

4.1 Dependency syntax tree

Dependency syntax analyzes a sentence into a dependency syntax tree to describe the dependencies between words, which is called dependency relation grammar [20]. Dependency syntax shows that there is a master-slave relationship between words. The dependency relationship matrix [21] can indicate a relationship between two words and how far the relationship is. Then, the closeness of the relationship between the two words is obtained so that the neural machine translation system can understand the general meaning of the sentence and obtain a more accurate translation result. This paper introduces the dependency syntax tree in syntactic analysis to integrate the structural information of the source language sentence into the input sequence and send it to the model for training.

For each source language to be translated, in order to more accurately understand the semantic information of the sentence, the Stanford CoreNLP toolkit is used to convert the source language sentence into the form of a dependency syntax tree, and then into the corresponding dependency matrix, because CBOW uses contextual The method of predicting the target word lacks the support of syntactic structure information. Based on the words in the context window, this paper selects the two words with the highest degree of closeness to the target word and adds them to the prediction sequence to obtain the word vector.

Taking "he told Tom to get his coat" as an example, Figure 2 shows the relevant dependency tree.

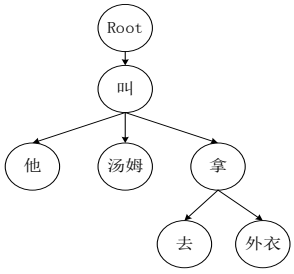

Fig. 2. Dependency syntax tree.

As shown in Figure 2, the distance between words in a sentence does not represent the closeness of the connection between the two words. The words that make up a sentence are also not arranged in order. There are long-distance dependencies between words. Therefore, it is necessary to use dependency syntax to extract semantic information in sentences in this paper. We can understand the structure of the sentence and the relationship between words through the extracted semantic information. However, the computer cannot directly identify the extracted information. This paper converts the dependency syntax tree into a dependency matrix. The definition of the dependency matrix is as follows:

1. The dependency weight is set to zero for two words with no dependencies.

2. For two words with a dependency relationship, as the distance between the two words in the dependency syntax tree increases, the weight of the dependency relationship should also decrease. As shown in formula (3).

$$D(i,j)=\begin{cases}0.5^{k-1}, \omega_j \text{ depands on } \omega_i \text{ in k lays} \\ 0, \omega_j \text{ does not depand on } \omega_i\end{cases} \tag{3}$$

The dependency matrix can clearly show the closeness of the relationship between words. The two words with the closest degree of dependency are added to the sequence used to predict the target word in the CBOW model to predict the target word to obtain the word vector containing the dependency information.

4.2 Long short-term memory network

The long short-term memory network is a recurrent neural network. Compared with the recurrent neural network, LSTM is special. It can be applied to longer sequences to learn long-term dependency information. LSTM has the cyclic structure of RNN, but unlike it, LSTM introduces three gate structures to realize the transfer of information in the network structure.

The role of LSTM in this paper is to obtain position encoding information and alleviate the limitations of position encoding in the original model. The LSTM model architecture is shown in Figure 3.

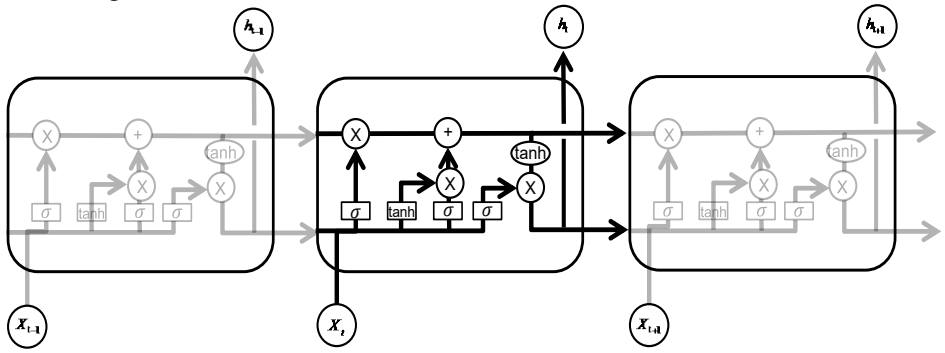

Fig. 3. LSTM model architecture.

This paper introduces the LSTM neural network structure to help obtain the position information of words in sentences. According to the memory characteristics of LSTM itself, the input sequence containing the dependencies between words and word vectors and word vectors is input into the LSTM model for training. One of the characteristics of LSTM is that it can solve the problem of long-distance dependence of sentences so that the output of each time step can be The information includes the content information of the current word and the previous word, so this paper uses the output of each time step of LSTM as the position information of the current word. Directly adding the position code and the word vector will blur the semantic representation of the word vector and cannot represent the position information well. Therefore, this paper adopts the concatnate method to fuse the position information and the word vector to improve the translation effect of the translation system. The improved overall model architecture in this paper is shown in Figure 4.

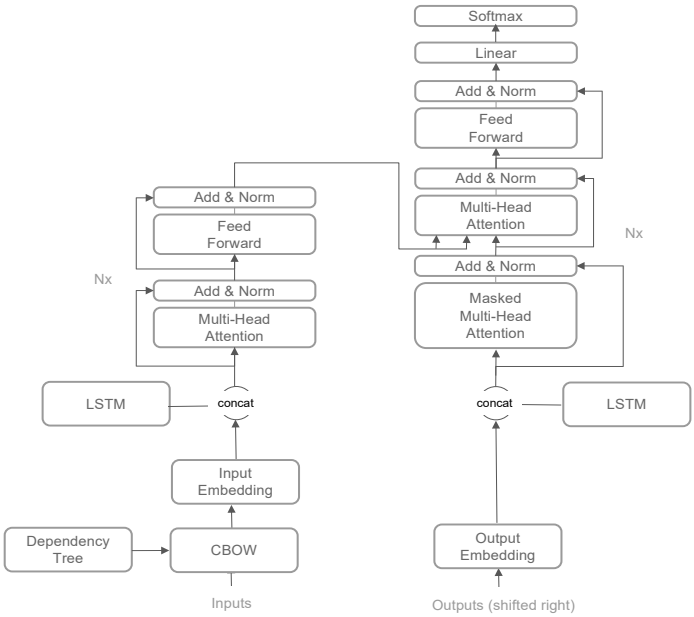

Fig. 4. DTCL model architecture.

5 Experimental setup

We uses the Transformer proposed by Ashish Vaswani et al. to improve the benchmark model and experimented on IWSLT14 by using Chinese-English language pairs.We use dev2010 as the validation set and test2010-2013 as the test set. This article uses the Stanford CoreNLP toolkit to obtain the dependency matrix of each sentence. The scale of the experimental data is shown in Table 1.

Table 1. Scale statistics of experimental data.

Language pair	Traing	Development	Test
zh-en	8.7M	43.4K	0.2M

Parameter setting: The basic structure of the model proposed in this paper is based on Transformer, using six encoder sub-layers and six decoder sub-layers to build the encoder and decoder of this model respectively. The word vector dimension of the model and the hidden layer size of the source and destination are both set to 512. The number of attention heads is 8, and the feedforward neural network dimension is 2048. The value of Dropout is set to 0.1, the initial value of the learning rate is set to 0.5, the model parameters are updated using the Adam algorithm [22], and the optimizer parameters are set to 1,2,3. Other related parameters are consistent with the transformer model proposed by Vaswani et al. Only sentences with no more than 100 source language sentences are used, and the context window size of the CBOW model is 2.

5.1 Results and analysis

In order to more intuitively understand the importance of the different improved modules of this model to the system performance, the proposed model is decomposed in the experiment process. The complete model proposed in this paper is compared with the model with a masked module to understand and analyze the importance of each sub-module to the

translation performance of the model. The experiment in this paper mainly analyzes the importance of the two sub-models added after the improvement of the original baseline model transformer, using the dependency matrix to add semantic information to the original model and using the LSTM model to obtain the positional encoding of the language sequence. For the module using the dependency tree, only the semantic relationship of the source language sentence is represented by the dependency relationship matrix, and the two words with the highest degree of closeness of each target word are added to the word vector in the prediction sequence. For the LSTM model, the input sequence obtained by the module using the dependency tree is input to the LSTM model for training, and the output of each time step is obtained. Because of the powerful memory function of LSTM, this paper uses the output of each time step as each word's location information in the sentence. This paper uses the BLEU value[23] as the judging criterion for the model. In this paper, DTCL (dependency tree combine LSTM) is used to represent the method proposed in this paper, "DTCL(-DT)" (DTCL without dependency tree) is used to represent the model that removes the dependency tree, and "DTCL(-L)" is used to represent the model that removes the LSTM.

The experimental results are shown in Table 2.

Table 2. Experimental results.

	Model	BLEU
Existing NMT Systems	GNMT[24]	27.24
	ConvS2S[12]	27.51
	Transformer	28.70
Our NMT Systems	DTCL(-L)	29.32
	DTCL(-DT)	29.54
	DTCL	29.93

As shown in Table 2, the translation model proposed in this paper outperforms other existing models in translation from Chinese to English. This paper uses dev2010 as the validation set and test2010-2013 as the test set to start the experiment. The average BLEU value of the model proposed in this paper can reach 29.93, which is 1.23 BLEU points higher than the benchmark model. It can be seen that the method proposed in this paper can improve the translation performance of the transformer model. In order to verify the effectiveness of the two sub-modules proposed in this paper, ablation experiments are performed in this paper. As shown in Table 1, the overall model proposed in this paper improves by 0.61 BLEU points than DTCL(-L), which indicates that adding semantic information to the model can better help the model understand the structural information of sentences, thereby improving translation performance. Furthermore, compared with DTCL(-DT), the model proposed in this paper improves 0.39 BLEU points. It shows that using the output of each time step of LSTM as the position information of the word can improve the translation accuracy of the translation system, thereby improving the translation performance of the translation system.

Figure 5 shows the change curve of the BLEU value of the Chinese to English translation on the validation set with the number of training steps. It can be seen from the figure that the method proposed in this paper has a good improvement in translation effect compared with the baseline model. It shows that the method proposed in this paper has a clearer understanding of the semantic results of sentences after adding linguistic knowledge to the baseline model. The position encoding information obtained by using LSTM makes the model proposed in this paper more accurate. Translate source language sentences, which produces more accurate translations.

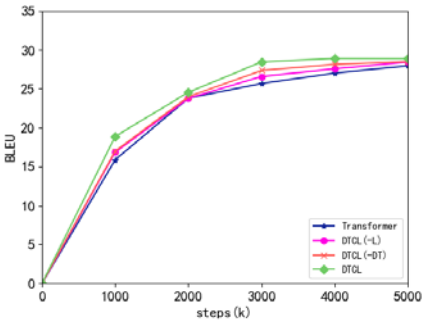

Fig. 5. The relationship between the number of iteration steps and the BLEU value of the model on the validation set.

6 Conclusion

This paper proposes a method to improve the accuracy of the input sequence in the DTCL model so that the input sequence can more accurately represent the source language. The dependency matrix is obtained using a dependency tree, and the word vectors of the two words with the highest degree of closeness of each target word are added to the prediction sequence. Positional encoding information, and integrate these two operations into the neural machine translation model Transformer to improve translation performance. Experiments show that our way of obtaining location information has an extensive performance improvement on the translation task of Chinese-English language pairs.

In the future work, I hope to continue to improve the performance of machine translation by improving the way to obtain sentence position information. Linguistic knowledge and neural networks are currently being introduced to improve the translation effect. The next step will be to change the structure of the attention model to obtain the position information of the sentence to further improve the translation performance of the model.

This work was financially supported by Special Foundation of Scientific and Technological Innovation for Young Scientists of Harbin, China (Grant No. 2017RAQXJ045). National Natural Science Foundation of China (NSFC) (Grant No. 61772160).

References

1. FENG Y, SHAO C Z. Frontiers in Neural Machine Translation: A Literature Review[J], Journal of Chinese information Processing,2020,34(07):1-18.
2. GAO M H, YU Z Q.A summary review of neural machine translation[J]. Journal of Yunnan Nationalities University (Natural Sciences Edition), 2019, 28(01):72-76.
3. LI Y C,XIONG D Y,ZHANG M.A survey of neural machine translation[J].Chinese Journal of Comput-ers,2018,41(12):2734-2755.
4. Kalchbrenner N, Blunsom P. Recurrent continuous translation models. 2013.
5. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.
6. Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations[J]. arXiv preprint arXiv:1803.02155, 2018.

7. LIU Y. Recent advances in neural machine teanslation[J].Journal of Computer Research and Development,2017,54(6):1144.
8. Eriguchi A, Hashimoto K , Tsuruoka Y . Tree-to-Sequence Attentional Neural Machine Translation[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016.
9. Robinson J J. Dependency structures and transformational rules[J]. Language, 1970: 259-285.
10. Mikolov T , Chen K , Corrado G , et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.
11. Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural computation, 1997, 9(8): 1735-1780.
12. Gehring J, Auli M, Grangier D, et al. Convolutional sequence to sequence learning[C]//International Conference on Machine Learning. PMLR, 2017: 1243-1252.
13. Wang X, Tu Z, Wang L, et al. Self-attention with structural position representations [J]. arXiv preprint arXiv:1909.00383, 2019.
14. Chen K, Wang R, Utiyama M, et al. Recurrent positional embedding for neural machine translation[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 1361-1367.
15. Liu X, Yu H F, Dhillon I, et al. Learning to encode position for transformer with continuous dynamical model[C]//International Conference on Machine Learning. PMLR, 2020: 6327-6335.
16. Li J, Xiong D, Tu Z, et al. Modeling source syntax for neural machine translation[J]. arXiv preprint arXiv:1705.01020, 2017.
17. Sennrich R, Haddow B. Linguistic input features improve neural machine translation [J]. arXiv preprint arXiv:1606.02892, 2016.
18. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
19. Ba J L, Kiros J R, Hinton G E. Layer normalization [J]. arXiv preprint arXiv:1607.06450, 2016.
20. AN J. Machine translation of long English sentence based on dependency parsing and sequence labeling [J].Journal of Lanzhou University of Technology,2018,1.
21. LI Z H. Research on Neural Machine Translation Combining Lexicology And Syntax[D].Shanghai Tiao Tong University,2019.
22. Kingma D P, Ba J. Adam: A method for stochastic optimization [J]. arXiv preprint arXiv:1412.6980, 2014.
23. Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002: 311-318.
24. Wu Y, Schuster M, Chen Z, et al. Google's neural machine translation system: Bridging the gap between human and machine translation [J]. arXiv preprint arXiv:1609.08144,2016.