 CMC, vol.59, no.2, pp.547-562, 2019

Dependency-Based Local Attention Approach to Neural Machine Translation

Jing Qiu[1], Yan Liu[2], Yuhan Chai[2], Yaqi Si[2], Shen Su[1,*], Le Wang[1,*] and Yue Wu[3]

Abstract: Recently dependency information has been used in different ways to improve neural machine translation. For example, add dependency labels to the hidden states of source words. Or the contiguous information of a source word would be found according to the dependency tree and then be learned independently and be added into Neural Machine Translation (NMT) model as a unit in various ways. However, these works are all limited to the use of dependency information to enrich the hidden states of source words. Since many works in Statistical Machine Translation (SMT) and NMT have proven the validity and potential of using dependency information. We believe that there are still many ways to apply dependency information in the NMT structure. In this paper, we explore a new way to use dependency information to improve NMT. Based on the theory of local attention mechanism, we present Dependency-based Local Attention Approach (DLAA), a new attention mechanism that allowed the NMT model to trace the dependency words related to the current translating words. Our work also indicates that dependency information could help to supervise attention mechanism. Experiment results on WMT 17 Chinese-to-English translation task shared training datasets show that our model is effective and perform distinctively on long sentence translation.

Keywords: Neural machine translation, attention mechanism, dependency parsing.

1 Introduction

Recently, Neural Machine Translation with attention-based encoder-decoder framework [Bahdanau, Cho and Bengio (2014)] has achieved state-of-the-art performances in many translation tasks. Typically, the encoder maps the necessary information of a source sentence into the corresponding hidden state vectors. According to the words currently being translated, these hidden state vectors are then assigned different weights by the attention

[1] Cyberspace Institute of Advanced Technology (CIAT) Guangzhou University, Guangzhou, 510006, China.

[2] Department of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, 050000, China.

[3] USC Information Sciences Institute, Marina del Rey, CA 90292, USA.

[*] Corresponding Author: Shen Su. Email: johnsuhit@gmail.com.

Le Wang. Email: wangle@gzhu.edu.cn.

mechanism. Finally, those weighted hidden state vectors are combined as a fixed length context vector that given in the decoder to generate translations. Therefore, enrich source sentences by various linguistic knowledge so that the encoder could learn more informative hidden state vectors is a hotspot direction of recent study. Among all linguistic knowledge, lexical knowledge, syntax, and semantics are three aspects that are currently prevalently applied in machine translation. As syntactic dependency trees can well represent dependency relationships between long-distance words among a sentence, there have been some works successfully introduced dependency information into NMT. Such as adding dependency label to each token of source sentences [Bojar, Chatterjee, Federmann et al. (2016)] or organizing related dependency information into a single unit for later use is all proven to be practicable [Chen, Wang, Utiyama et al. (2017)]. There is also some work, such as [Wu, Zhou and Zhang (2017)], independently learning dependency information to generate dependency hidden state vectors by increase another encoder.

However, the method of boosting the encoder-decoder framework by adding a lot of extra information to the encoder side may put the additional burden to the model itself. For example, the computational complexity may be increased. As stated in Chen et al. [Chen, Wang, Utiyama et al. (2017)], their model is 25% slower than the compared standard NMT model. We assume another potential problem is dependency information doesn't be used adequate, for just simply joint the dependency information in the encoder side.

In this paper, we propose a novel attention approach. We consider that while enriching source sentences that let the encoder could learn more informative information is very important in the encoder-decoder framework, however, attention mechanism in the decoder side is the most efficient part that influences the framework to generate the correct translation. Therefore, we present Dependency-based Local Attention Approach (DLAA), a new type of attention mechanism to improve NMT. DLAA based on the theory of local attention mechanism. Levering the dependency information influence the attention mechanism to retrospect on the source words that semantic related to current translating (Section 5). In this way, not only long-distance words that in terms of current translating could be captured, a more accurate translation model could be trained by rationally explore the extra semantic or syntactic information.

Experimentally, we prove that our approach is effective in the translation task between Chinese and English. Results show that our approach worked effectively and performed distinctively on long sentence translation.

2 Related work

As the modeling formulation of neural machine translation encoder-decoder framework is overly simplistic [Cohn, Hoang, Vymolova et al. (2016)] and in terms of the alignment accuracy, attention-based NMT model is not as good as the conventional statistics alignment model [Liu, Utiyama, Finch et al. (2016)]. Therefore, under those considerations, many attempts have been tried to make an improvement.

Merging knowledge of linguistics [Wang, Wang, Guo et al. (2018)] has been proved to

be valid to improve the performance of machine translation [Li, Resnik and Daumé III (2013)]. Integrating syntactic information becomes a trend for it has the advantage in capturing information that steps across in a long distance. From this perspective, Li et al. [Li, Xiong, Tu et al. (2017)] linearize a phrase tree into a structural label sequence and utilize another RNN to model these labels. Then the hidden vectors of parse tree labels and source words have been tried to combine in three different ways to improve the translation accuracy of NMT. Wu et al. [Wu, Zhou and Zhang (2017)] increased another two RNN to take advantage of the dependency tree to explicitly model source word. Dependency structures are extracted from the dependency tree in two way to enrich source word. Child Enriched Structure RNN (CES-RNN) that enrich source child nodes with global syntactic information and Head Enriched Structure RNN(HES-RNN) to enrich source head nodes with its child nodes. Therefore, each source node could contain relatively comprehensive information.

Besides the straightforward way to model syntactic information by sequence network RNN, other classes of neural networks which is more suitable to modeling graph-structured data are also be exploited, as syntactic information is always contained with edges and nodes. In Bastings et al. [Bastings, Titov, Aziz et al. (2017)], they employed graph-convolution network (GCN) on top of a normal encoder network to combined information of dependency trees. GCN is a neural network which contains multiple layers that directly modeling information on the graph, information about syntactic neighborhoods of source words could be directly modeled through this special kind of network. The work Marcheggiani et al. [Marcheggiani and Titov (2017)] also verified GCN is effective for NLP tasks.

Both above methods modeling syntactic knowledge in the encoder side, however, the decoder side is also very important. As the point raised in Tu et al. [Tu, Liu, Lu et al. (2017)], they find the source contexts impact on translation adequacy while target contexts affect translation fluency. Thence, some works started to focus on improving the decoder side. The method Sequence-to-Dependency NMT (SD-NMT) [Wu, Zhang, Yang et al. (2017)] was proposed to face the challenge. In this method, dependency structure was dynamically constructed in consist with the process of generating target word. Letting a single neural network have the capability of performing target word generation and syntactic structure construction simultaneously. And the resulting dependency tree largely influences the generation of translation at the current moment.

Since attention mechanism is a weighted part in NMT, Chen et al. [Chen, Huang, Chiang et al. (2017)] applied the source syntax into the attention part to enhance the alignment accuracy. Specifically, the coverage model was employed in their work by added the coverage vector for each node [Tu, Lu, Liu et al. (2016)], along with this method, the child nodes information was adopted in the coverage vector, and then the coverage vector was made use of updating the attention.

Another kind of attempt for using syntactic knowledge is raised for the consideration that each kind of parse tree generated by parsers contains errors itself. Zaremoodi et al. [Zaremoodi and Haffari (2018)] proposed forest-to-sequence attentional NMT model, based on

the tree-to-sequence model method [Eriguchi, Hashimoto and Tsuruoka (2016)], which inherit another RNN to model hierarchical syntactic information. Different from tree-to-sequence use only one parse tree, they use packed forests which contains different kinds of parse trees.

This work also draws on the idea of the Big Data Learning [Han, Tian, Huang et al. (2018)], Data-driven model [Tian, Su, Shi et al. (2019); Qiu, Chai, Liu et al. (2018)], Cloud System [Li, Sun, Jiang et al. (2018)], Internet of Things [Chen, Tian, Cui et al. (2018)].

3 Background

In this section, we mainly introduce the following aspects. The knowledge of dependency parsing is briefly introduced in part 3.1. Then we introduce the standard attention-based NMT model proposed by Vinyals et al. [Vinyals, Kaiser, Koo et al. (2015)] in part 3.2. Progressively, the local attention mechanism which improved on standard attention mechanism (global mechanism) is introduced in part 3.3 [Luong, Pham and Manning (2015)]. Both models consists of an encoder and a decoder. Finally, we introduce a recent work which also explored dependency information, as one of the comparisons of our model. Both models consist of the encoder-decoder framework.

3.1 Introduction of dependency parsing

The knowledge of dependency parsing or dependency grammar focuses on the relationship between a word and another word among a sentence. Dependency is a binary asymmetric relation between a central word and its subordinates [Bird, Klein and Loper (2009)]. The central word of a sentence is usually taken to be the tensed verb, and all other words either depend on the central word directly or associated with the central word through a dependency path indirectly.

The dependency parsing graph is usually represented by a labeled directed graph. Among the graph, words are represented as nodes, the dependency relationship between the central word and its subordinate is represented as the tagged arc. For example, as is shown in Fig. 1, "root" represents that the central word is "chi" (eating), although "wan" (playing) is another important verb among the sentences, the dependency parsing tool we used correctly tagged their relationship as "conj", which means the verb "chi" and the verb "wan" is two parallel words. The example of the relationship of the central word and its subordinate is "chi" and "pingguo" (apple), the dependency parsing result shows their relationship is the direct object "dobj".

The more important meaning of dependency parsing is reflected in the two words "pingguo" that arises in the sentence. Among the sentence, the first "pingguo" (apple) means fruit apple, and the second "pinguo" (Apple) represents the name of a company. As we can see in Tab. 1, although the two words are identical in character form, while the meaning of them varies greatly, and should not both translate them in "apple".

By adding the dependency constraint, we can see it shows that the second "pingguo" has a direct dependency relationship with "shouji" (cellphone), therefore, in theory, the RNN

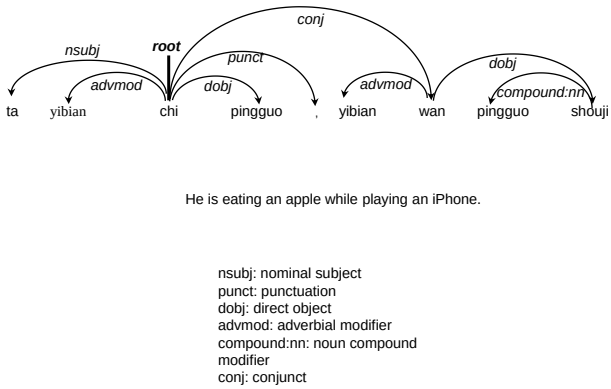

Figure 1: The example of parsing the sentence "ta yibian chi pingguo, yibian wan pingguo shouji.", the corresponding english is "He is eating an apple while playing an iPhone."

Table 1: A translation example demonstrates the space for NMT to improve

Source	ta yibian chi pingguo , yibian wan pingguo shouji.
Correct translate	He is eating an apple while playing an iPhone.
NMT translate example	He is eating an apple while playing an Apple phone.

could learn the difference between the two identical words "pingguo" and give different hidden states to each word. Under these points, the model has a greater chance to translate the second "pingguo" and its following word "shouji" into "iPhone", which is the correct translation.

Through this example, we can see that although the neural network can automatically learn the characteristics of the translation task, due to the limitations of the corpus or the current neural network architecture, enhancing the semantic information through the dependency syntax can help us train the neural network translation model more accurately.

In addition, although the dependency parser still makes some mistakes, while the parser which driven by the neural network model has improved the accuracy a lot [Chen and Manning (2014)], so the dependency information could be proficiently used in the translation task.

3.2 Neural machine translation with standard attention mechanism

Usually, a source input sentence is firstly tokenized as $x_j \in (x_1, ..., x_J)$ and then each token is embedded as a vector V_{x_j}, as shown in Fig. 2. After that, the encoder encodes

those source vectors into a sequence of hidden state vectors:

$$h_j^e = f\left(V_{x_j}, h_{j-1}^e\right) \tag{1}$$

where h_j^e is an encoder hidden state vector that generated by a Recurrent Neural Network (RNN) f. Our work used Long Short-Term Memory (LSTM) neural network as f, for instance.

The decoder is often trained to compute the probability of next target word y_t by a softmax layer q:

$$p\left(y_t|y_{<t}\right) = q\left(\hat{y}_{t-1}, h_t^d, c_t\right) \tag{2}$$

where $\hat{y}_{t-1}$ is the embedding vector of previously yielding translation word, and h_t^d is a current time decoder hidden state vector generated by an RNN g:

$$h_t^d = g\left(\hat{y}_{t-1}, h_{t-1}^d, c_t\right) \tag{3}$$

among the last two equations, c_t is the context vector in terms of current translating, which is computed as a weighted sum of all the encoder hidden states:

$$c_t = \sum_{j=1}^{J} \alpha_{tj} h_j^e \tag{4}$$

where the alignment weight α_{tj} of each encoder hidden state h_j^e is computed as:

$$\alpha_{tj} = \frac{exp\left(e_{tj}\right)}{\sum_{k=1}^{J} exp\left(e_{tk}\right)} \tag{5}$$

where e_{tj} is an alignment model which scores how well the inputs around position j and the output at the current time t match:

$$e_{tj} = s\left(h_{t-1}^d, h_j^e\right) \tag{6}$$

where s is the score function that has different alternatives.

3.3 Local attention mechanism

Taking all the encoder hidden state vectors into account when deriving the context vector of the current time is the method of the traditional attention approach, also known as the global attention mechanism. Considering the computational expensive and impractical for global attention to translating long sentences, the theory of local attention mechanism was proposed by Luong et al. [Luong, Pham and Manning (2015)]. This theory selectively focuses only on a small subset of the encoder hidden states in terms of per target word, has an advantage of avoiding the expensive computation and training easily than the traditional global attention approach.

The main implementation idea of local attention mechanism is to select a position within the length of the source sentence before generating each context vector. Centering on

this position, set a fixed size window and there will be some source hidden state vectors included in the window. Finally, only the hidden state vectors contain in the window are selected and participate in generating the current context vector.

This work proposed two methods to select the position and therefore develops two types of local attention approach model.

One way is monotonic alignment model: This model just simply set the position equal to the current time step for assuming that source and target sequences are roughly monotonically aligned. Another way is predictive alignment model: By means of an independent network, this model learns to predict an alignment position.

Specifically, in both of the two methods, the context vector c_t is now a weighted sum of encoder hidden states which only included within a window $[p_t - D, p_t + D]$:

$$c_t = \sum_{j}^{j \in [p_t - D, p_t + D]} \alpha_{tj} h_j^e \tag{7}$$

D is the half size of the window and is empirically selected. p_t equals to current time t when using monotonic alignment model. When using predictive alignment model, p_t is an aligned position generated by the model according to the following equation:

$$p_t = S \cdot sigmoid\left(v_p^\top tanh\left(W_p h_t^d\right)\right) \tag{8}$$

where W_p and v_p are the hyper parameters to be learned. S is the length of source sentence. As the result of $sigmoid$, $p_t \in [0, S]$.

3.4 Neural machine translation with source dependency representation

In this section, we will introduce a work [Chen, Wang, Utiyama et al. (2017)] in a more detail way which also exploited dependency information to improve NMT model. Part of our work to improve the NMT model is inspired by this article. And according to the idea of the article, we try our best to reimplement their models as the comparison of our model.

The work proposed two types of models: SDRNMT-1 (Neural machine translation with source dependency representation) and SDRNMT-2 to exploited the efficient way to use dependency information.

Different from the previous work that simply combined the labels of dependency information to source sentences, this work uses a relatively complicated way, that is, using an independent neural network to learn the dependency information. The learned information is then combined with the NMT model in different ways.

The first step is the extraction and organization of dependency information. In their work, a dependency unit was extracted for each source word x_j from the dependency tree. The dependency unit is organized as the following:

$$U_j = \left\langle PA_{x_j}, SI_{x_j}, CH_{x_j} \right\rangle \tag{9}$$

where U_j represents the dependency unit of x_j; PA_{x_j}, SI_{x_j}, CH_{x_j} denotes the parent, siblings, children words of x_j respectively in a sentence tree.

Then a simple Convolution Neural Network (CNN) was designed to learn the Source Dependency Representation (SDR) for each of the organized dependency units.

Therefore, compared with the standard attention-based NMT model, the encoder of the two models: SDRNMT-1 and SDRNMT-2 all consist of a convolutional architecture and an RNN. In this way, the large size of dependency units with sparsity issues was tackled and a compositional representation of dependency information was learned.

The innovation of model SDRNMT-1 is leverage the dependency information (SDR) and the source word embedding vector together to generate the source hidden state vectors:

$$h_j^e = f\left(V_{x_j} : V_{U_j}, h_{j-1}^e\right) \tag{10}$$

where V_{U_j} is the vector denotes of SDR, and ":" denotes the operation of vectors concatenation.

The remaining architecture of SDRNMT-1 is the same as the standard attention-based NMT model.

Unlike model SDRNMT-1, which only uses dependency information on the encoder side, model SDRNMT-2 makes dependency information participate in various parts of the encoder-decoder framework.

Instead of concatenating source word embedding and SDR together, SDRNMT-2 let SDR be an independent part to generate its own hidden state vectors:

$$d_j^e = q\left(V_{U_j}, d_{j-1}^e\right) \tag{11}$$

where q is the independent RNN to learn SDR hidden state vectors. SDRNMT-2 also generate the separate context vectors for SDR hidden states:

$$c_t^d = \sum_{j=1}^{J} \widetilde{\alpha}_{tj} d_j^e \tag{12}$$

at the same time, the context vector of source hidden states is:

$$c_t^e = \sum_{j=1}^{J} \widetilde{\alpha}_{tj} h_j^e \tag{13}$$

where $\widetilde{\alpha}_{tj}$ is a new alignment weight that made a further process of the separate alignment weights (source hidden states and dependency hidden states) by adding a hyperparameter to control the importance of the two part.

Now the current target hidden state vectors of dependency information and source word is computed as:

$$h_t^d = g\left(\hat{y}_{t-1}, h_{t-1}^d, c_t^e\right) \tag{14}$$

$$h_t^{dep} = g\left(\hat{y}_{t-1}, h_{t-1}^{dep}, c_t^d\right) \tag{15}$$

Now, the work arranged the probability of next target word is:

$$p\left(y_t | y_{<t}\right) = q\left(\hat{y}_{t-1}, h_t^d, h_t^{dep}, c_t^e, c_t^d\right) \tag{16}$$

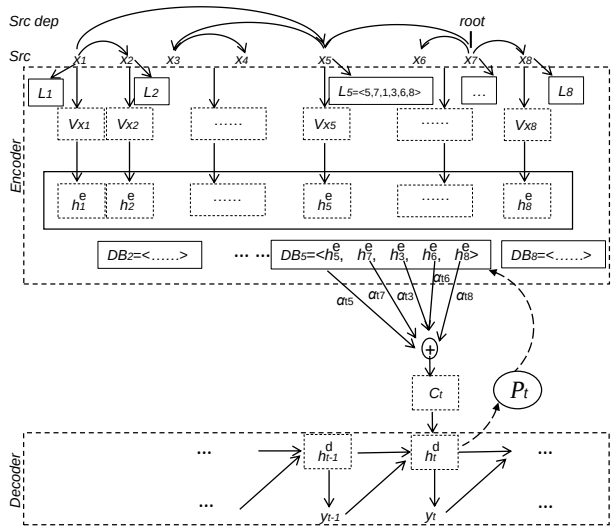

Figure 2: NMT with local dependency attention approach

4 Organizing dependency information

Inspired by Chen et al. [Chen, Huang, Chiang et al. (2017)], we introduced above, which exploiting dependency information from the dependency tree as a unit to increases extra information for each source word, we organized our dependency information unit L_j for each source word as the following:

$$L_j = \left\langle L_{x_j}, L_{PA_{x_j}}, L_{CH_{x_j}}, L_{SI_{x_j}} \right\rangle \tag{17}$$

different with the way directly organizing dependency words itself as a unit, we record the location of the words in a sentence and organize it as a unit. x_j is one of the source token, PA_{x_j}, SI_{x_j}, CH_{x_j} denotes the parent, siblings, children words of x_j respectively in a sentence tree. L_{x_j} represents the location of x_j itself, where $L_{PA_{x_j}}, L_{CH_{x_j}}, L_{SI_{x_j}}$ denotes the location of parent, children and siblings words of x_j respectively. Take x_5 in Fig. 2 as an example, the solid box represents L_5: $L_{x_5} = \langle 5 \rangle$, $L_{PA_{x_j}} = \langle 7 \rangle$, $L_{CH_{x_2}} = \langle 1, 3 \rangle$, $L_{SI_{x_2}} = \langle 6, 8 \rangle$, that is, $L_2 = \langle 5, 7, 1, 3, 6, 8 \rangle$. Empirically, we constrained the number of location information in a unit is ten, which adopt nine dependency words of a source token. Specifically, most token contains no more than nine dependency words, we have tried to pad L_j which shorter than ten with "/", but experiment results show that this way of padding is insufficient and computational wasteful. Therefore, we padded the spare information in L_j with the location of the words which around x_j.

5 Neural machine translation with dependency-based local attention approach

In order to address the potential issues which narrated in section one, we proposed DLAA (dependency-based local attention approach). A new type of attention approach to enhance NMT.

In our model, the encoder part and the decoder part are same with the traditional standard attentional NMT, which, implemented by RNNs. However, the inputs of the encoder, in addition to the tokens of source sentence, its corresponding location information unit of each token is also included, as shown in Fig. 2. After the tokens of source inputs are embedded and represent as the encoder hidden states, dependency blocks of each token were generated, by using the location information that contains in L_j. In detail, the dependency block is defined as the following:

$$DB_j = \left\langle h_j^e, \left\langle PA_{h_j^e} \right\rangle, \left\langle CH_{h_j^e} \right\rangle, \left\langle SI_{h_j^e} \right\rangle \right\rangle \tag{18}$$

among the equation, h_j^e is the encoder hidden state of x_j itself, $\left\langle PA_{h_j^e} \right\rangle$ is the encoder hidden states of parent of x_j. Similar to $\left\langle PA_{h_j^e} \right\rangle$, $\left\langle CH_{h_j^e} \right\rangle$ and $\left\langle SI_{h_j^e} \right\rangle$ is the corresponding encoder hidden states of x_j.

After generating the dependency blocks, one of them was selected in line with the generated aligned position p_t, according to the Eq. (8).

The theory of local attention mechanism chooses to focus only on a small subset encoder hidden states during the attention compute process, while DLAA chooses to focuses on those encoder hidden states that contain in the dependency block:

$$c_t = \sum_{j}^{j \in DB_j} \alpha_{tj} h_j^e \tag{19}$$

Compared with local attention mechanism only focuses a fixed subset of encoder hidden states around the choose position, DLAA chooses those encoder hidden states that have semantic relationships with current chooses position, in this way, information has a relationship with current time but distance long could also be captured.

6 Experiment

6.1 Setting up

We carried out our experiments on Chinese-to-English translation and conducted three sets of experiments respectively. The datasets are both extracted from WMT17 translation task shared corpora. Experiment one included 0.23 M training sentence pairs extracted from news-commentary [Bojar, Chatterjee, Federmann et al. (2017)]. The validation dataset and test datasets are extracted from the corpus as well. Training dataset of experiment two and experiment three both extracted from The United Nations Parallel corpus [Bojar, Chatterjee, Federmann et al. (2017)], included 0.9 M and 2 M sentence pairs separately. Their validation dataset and test datasets are extracted from the corpus itself as well. Specifically,

we group our test dataset by sentence length. For example, "30" indicates that the length of the sentences is between 20 and 30. Each group of the test dataset contains a thousand sentences except the group "70' and "80", for long sentences in such a length is rear in the corpus. The dependency tree for each Chinese sentence is generated by the Stanford CoreNLP [Manning, Surdeanu, Bauer et al. (2014)]. The processing speed in 6 G memory is about 0.3 M per hour. Translation quality was evaluated by case-insensitive BLEU-4 [Papineni, Roukos, Ward et al. (2002)] metric.

We use the sequence to sequence model implemented by the NMT tutorial Luong, Brevdo and Zhao (2017) of Tensorflow, with its default settings as one of our baseline system.

Other models used as comparative experiments are local predictive alignment model, SDRNMT-1 and SDRNMT-2. In order to be consistent with the number of dependency word that set, SDRNMT-1 and SDRNMT-2 both retain 10 dependency word. The window size of the local predictive alignment model is also 10.

We have tried our best to re-implement model SDRNMT-1 and SDRNMT-2. Since the platform we re-implement on is TensorFlow, the implementation of the convolutional neural network is slightly different with the original.

6.2 Training

The Chinese and English vocabularies are all limited in 40 K for our model and the baseline model. Other words are replaced by the special symbol "UNK". The maximum training length of Chinese sentences is 40 dues to the equipment limitation.

For the comparison of traditional standard attention-based NMT model and Dependency-based local attention approach NMT model, each RNN layer contains 1024 hidden units. The word embeddings are 1024 dimensional. A batch of size 64 stochastic gradient descent (SGD) was used to train the networks.

Due to the limitations of experimental conditions, for the comparison of DLAA NMT model and local predictive alignment model, SDRNMT-1, SDRNMT-2, each RNN layer contains 620 hidden units. The word embeddings are 620 dimensional. A batch of size 32 stochastic gradient descent (SGD) was used to train the networks.

6.3 Results and analyses

Tab. 2, Tab. 3 and Tab. 4 lists the results conducted on the three datasets. From the average indicator, we observe that our approach indeed improves the translation quality of the tradition attentional NMT system. This indicates that our way of using dependency information is effective and in the right way. However, as shown in both tables, our approach performs not ideal on short sentences, we consume that the reduced hidden states for attention mechanism to attend hurt the performance of NMT when translating short sentences. But for long sentences, the reduced information guided by dependency information is still effective for improving the performance of NMT. On the other hand, the results also certificate that sufficient source context is important for the NMT system.

Table 2: Experiment on 0.23 M traning dataset

System	10	20	30	40	50	60	70	80	average
Traditional	9.52	6.23	4.14	3.55	2.85	2.28	1.52	1.76	3.98
DLAA	8.96	6.41	4.16	3.39	2.92	2.45	1.87	1.99	4.01
Difference	-0.56	+0.18	+0.02	-0.16	+0.07	+0.17	+0.35	+0.23	+0.03

Table 3: Experiment on 0.9 M traning dataset

System	10	20	30	40	50	60	70	80	average
Traditional	37.35	25.85	15.31	13.57	10.91	11.02	10.09	10.39	16.81
DLAA	36.99	25.19	16.00	13.72	11.07	11.08	9.33	13.54	17.12
Difference	-0.36	-0.66	+0.69	-0.03	+0.16	+0.06	+0.35	-0.76	+0.31

Table 4: Experiment on 2 M traning dataset

System	10	20	30	40	50	60	70	80	average
Traditional	42.32	26.39	20.04	15.89	14.07	12.94	13.09	11.91	19.58
DLAA	40.76	27.37	21.32	17.29	14.88	13.41	12.87	11.98	19.98
Difference	-1.56	+0.98	+1.28	+0.14	+0.81	+0.47	-0.22	+0.07	+0.40

Tab. 5 shows the comparison results of model DLAA NMT, Predictive alignment NMT, SDRNMT-1, SDRNMT-2 carried on training dataset 2 M. Although we have tried our best to re-implement SDRNMT-1 and SDRNMT-2, it shows less effective than our model and the local predictive alignment model, perhaps for the reason that we didn't use the training technique such as "dropout" to make the model achieve its best states. Although our model did not show excess performance than the local predictive alignment model, it also shows its competitiveness in translate long sentences.

Table 5: Comparison experiments on 2 M traning dataset of DLAA, local predictive alignment model (local_P), SDRNMT-1, SDRNMT-2

System	10	20	30	40	50	60	70	80	Average
DLAA	40.48	24.46	17.73	14.42	13.07	11.99	11.55	10.84	18.07
Local_P	40.33	24.86	18.36	15.21	12.79	12.05	11.79	10.81	18.28
SDRNMT-1	37.23	21.40	15.75	12.82	11.52	11.11	10.75	10.04	16.33
SDRNMT-2	37.09	21.48	15.39	11.69	10.85	10.21	10.31	9.29	15.79

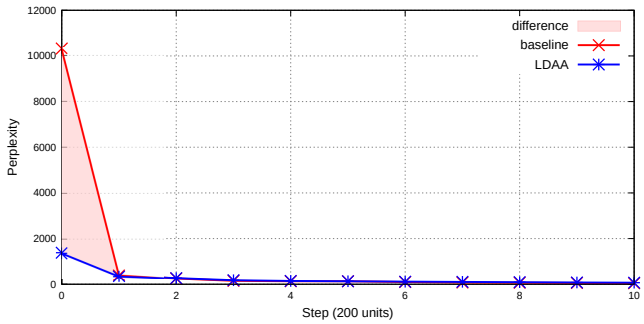

Figure 3: Part perplexity performance of NMT with LDAA and conventional NMT, the difference shows that NMT with LDAA learns more effective information to train the NMT model

6.4 Analyses of perplexity

Perplexity is a commonly used evaluation indicator of the language model. In simple terms, the language model is a model used to calculate the probability of a sentence, that is, the probability of adjudicating whether a sentence belongs to human language habits. For example, a given sentence is represented as:

$$S = s_1, s_2, ..., s_m \tag{20}$$

where $s_1, s_2, ...,$ is the words consisted of the sentences. The probability of the sentence can be expressed as:

$$P(S) = P(s_1, s_2, ..., s_m) = P(s_1) P(s_2|s_1) ...P(s_m|s_1, s_2, ..., s_{m-1}) \tag{21}$$

Given that the previous M words, the conditional probability of the M+1th word is modeled, that is, we hope that the language model could predict the M+1th word.

The basic idea of Perplexity is, the higher the probability value that the language model given to the sentences on test dataset, the better the model is. As we know, the sentences on the test dataset are both normal sentences. The formula of Perplexity is as follows:

$$Perplexity(S) = P(s_1, s_2, ..., s_m)^{-\frac{1}{M}} = \sqrt[M]{\frac{1}{P(s_1, s_2, ..., s_m)}} \tag{22}$$

Know by the formula, the smaller the perplexity is, the better the language model to generate sentences with high probability.

Fig. 3 has shown the part perplexity during the training, the NMT model equipped with DLAA has a much smaller perplexity value than the normal NMT model at the very beginning, which indicates that our model is very protentional in modeling the language model in a fast and efficient way. Although at each subsequent step, our perplexity values converge a little slower than the normal NMT. But finally, they arrived the same converge value.

7 Conclusion and future work

In this paper, we proposed a new attention approach DLAA to improve the translation performance of the NMT system based on the theory of local attention mechanism. Syntactic knowledge dependency information was used to mine deep relationships between words in a sentence to insurance the translation quality. Experiments on Chinese-to-English translation tasks show that our approach is effective and improve the translation performance of the conventional NMT system. While for the problems presented in our experiments need a further exploration. We will also compare our work with newer NMT models.

As syntactic knowledge has been proved to be useful in traditional statistical machine translation, we believe it could also help to improve NMT. A lot of works has proved so. In the further, we plan to explore more efficient ways to use syntactic knowledge and fix the problems represented in current work.

Acknowledgement: This research was funded in part by the National Natural Science Foundation of China (61871140, 61872100, 61572153, U1636215, 61572492, 61672020), the National Key research and Development Plan (Grant No. 2018YFB0803504), and Open Fund of Beijing Key Laboratory of IOT Information Security Technology (J6V0011104).

References

Bahdanau, D.; Cho, K.; Bengio, Y. (2014): Neural machine translation by jointly learning to align and translate. arxiv:1409.0473.

Bastings, J.; Titov, I.; Aziz, W.; Marcheggiani, D.; Sima'an, K. (2017): Graph convolutional encoders for syntax-aware neural machine translation. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1957-1967.

Bird, S.; Klein, E.; Loper, E. (2009): *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Bojar, O.; Chatterjee, R.; Federmann, C.; Graham, Y.; Haddow, B. et al. (2016): Findings of the 2016 conference on machine translation. *ACL 2016 First Conference on Machine Translation*, pp. 131-198.

Bojar, O.; Chatterjee, R.; Federmann, C.; Graham, Y.; Haddow, B. et al. (2017): Findings of the 2017 conference on machine translation. *Proceedings of the Second Conference on Machine Translation*, vol. 2, pp. 169-214.

Chen, D.; Manning, C. (2014): A fast and accurate dependency parser using neural networks. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 740-750.

Chen, H.; Huang, S.; Chiang, D.; Chen, J. (2017): Improved neural machine translation with a syntax-aware encoder and decoder. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1936-1945.

Chen, J.; Tian, Z.; Cui, X.; Yin, L.; Wang, X. (2018): Trust architecture and reputation evaluation for internet of things. *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-9.

Chen, K.; Wang, R.; Utiyama, M.; Liu, L.; Tamura, A. et al. (2017): Neural machine translation with source dependency representation. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2846-2852.

Cohn, T.; Hoang, C. D. V.; Vymolova, E.; Yao, K.; Dyer, C. et al. (2016): Incorporating structural alignment biases into an attentional neural translation model. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 876-885.

Eriguchi, A.; Hashimoto, K.; Tsuruoka, Y. (2016): Tree-to-sequence attentional neural machine translation. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 823-833.

Han, W.; Tian, Z.; Huang, Z.; Li, S.; Jia, Y. (2018): Bidirectional self-adaptive resampling in internet of things big data learning. *Multimedia Tools and Applications*, pp. 1-16.

Li, J.; Resnik, P.; Daumé III, H. (2013): Modeling syntactic and semantic structures in hierarchical phrase-based translation. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 540-549.

Li, J.; Xiong, D.; Tu, Z.; Zhu, M.; Zhang, M. et al. (2017): Modeling source syntax for neural machine translation. arxiv:1705.01020.

Li, M.; Sun, Y.; Jiang, Y.; Tian, Z. (2018): Answering the min-cost quality-aware query on multi-sources in sensor-cloud systems. *Sensors*, vol. 18, no. 12, pp. 4486.

Liu, L.; Utiyama, M.; Finch, A.; Sumita, E. (2016): Neural machine translation with supervised attention. arxiv:1609.04186.

Luong, M.; Brevdo, E.; Zhao, R. (2017): Neural machine translation (seq2seq) tutorial. https://github.com/tensorflow/nmt.

Luong, M. T.; Pham, H.; Manning, C. D. (2015): Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412-1421.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S. J. et al. (2014): The Stanford CoreNLP natural language processing toolkit. *Association for Computational Linguistics (ACL) System Demonstrations*, pp. 55-60.

Marcheggiani, D.; Titov, I. (2017): Encoding sentences with graph convolutional networks for semantic role labeling. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1506-1515.

Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. J. (2002): Bleu: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 311-318.

Qiu, J.; Chai, Y.; Liu, Y.; Gu, Z.; Li, S. et al. (2018): Automatic non-taxonomic relation extraction from big data in smart city. *IEEE Access*, vol. 6, pp. 74854-74864.

Tian, Z.; Su, S.; Shi, W.; Du, X.; Guizani, M. et al. (2019): A data-driven method for future internet route decision modeling. *Future Generation Computer Systems*.

Tu, Z.; Liu, Y.; Lu, Z.; Liu, X.; Li, H. (2017): Context gates for neural machine translation. *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 87-99.

Tu, Z.; Lu, Z.; Liu, Y.; Liu, X.; Li, H. (2016): Modeling coverage for neural machine translation. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 76-85.

Vinyals, O.; Kaiser, Ł.; Koo, T.; Petrov, S.; Sutskever, I. et al. (2015): Grammar as a foreign language. *Advances in Neural Information Processing Systems*, pp. 2773-2781.

Wang, M.; Wang, J.; Guo, L.; Harn, L. (2018): Inverted xml access control model based on ontology semantic dependency. *Computers, Materials & Continua*, vol. 55, no. 3, pp. 465-482.

Wu, S.; Zhang, D.; Yang, N.; Li, M.; Zhou, M. (2017): Sequence-to-dependency neural machine translation. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 698-707.

Wu, S.; Zhou, M.; Zhang, D. (2017): Improved neural machine translation with source syntax. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 4179-4185.

Zaremoodi, P.; Haffari, G. (2018): Incorporating syntactic uncertainty in neural machine translation with a forest-to-sequence model. *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1421-1429.