Neural Machine Translation with Attention Based on a New Syntactic Branch Distance

Ru Peng[1], Zhitao Chen[1], Tianyong Hao[2(✉)], and Yi Fang[1(✉)]

[1] School of Information Engineering, Guangdong University of Technology, Guangzhou, China
pengru709909347@gmail.com, chenzhitao@mail2.gdut.edu.cn, fangyi@gdut.edu.cn
[2] School of Computer Science, South China Normal University, Guangzhou, China
haoty@m.scnu.edu.cn

Abstract. Attention mechanism has been proved to be able to improve the quality of neural machine translation by selectively focusing on partial words of a source sentence during translation process. Attention mechanism usually focuses on local attention by using solely the linear index distance of words while ignores syntax structures of sentences. In this paper, we extend local attention through syntax distance constraint, and propose an attention mechanism based on a new syntactic branch distance, which simultaneously pays attention to words with similar linear index distances and syntax-related words. Based on the English-to-German translation task, experiment results showed that our model outperforms a recent baseline method with an improvement of 1.61 BLEU points, demonstrating the effectiveness of the proposed model.

Keywords: Neural machine translation · Attention mechanism · Syntactic branch distance · Syntax structure

1 Introduction

In the past few years, Neural Machine Translation (NMT) has made rapid progresses, showing superior performance compared to traditional statistical machine translation [1–3]. Many researchers have conducted extensive research on neural networks and attention mechanisms in NMT, which has promoted the rapid development of machine translation. Attention mechanism is critical to improve the translation performance of sentences in NMT. The research about attention mechanism has been in full swing. Bahdanau et al. [4] proposed an attentional NMT model (called global attention), which dynamically capture every contexts of source sentences in each decoding step, improving the performance of the NMT. Luong et al. [5] further refined global attention into local attention, selectively focusing source context of the fixed window size in each decoding step, and experimentally proved its effectiveness in German-to-English and English-to-German translation tasks. However, traditional attention mechanism, such as global attention [4] and local attention [5], only focuses on the sequential structure of sentences and ignores the dependencies between words. This does not

S. Huang and K. Knight (Eds.): CCMT 2019, CCIS 1104, pp. 47–57, 2019.
https://doi.org/10.1007/978-981-15-1721-1_5

conform to the rules of syntactic analysis, which may lead to some common syntax errors and affect the quality of the sentence translation.

In order to address the above problems, we propose a new attention mechanism based on the syntactic dependency tree of sentences. It simultaneously focuses on the sequential structure and syntactic structure of sentences for reducing the noise brought by grammar trees to some extent. In this paper, we propose a new syntactic branch distance constraint to extend local attention, predicting the encoder state associated with source words syntactically relating to target words. According to the dependency tree of a source sentence, a more effective context vector is calculated according to the syntactic branch distance for predicting target words. Experiments on the ISWLT2017 EN-DE translation task, our model is compared with a recent baseline method and the results show that our model improves 1.28 BLEU points over the baseline method.

2 Related Work

2.1 Syntax Representation for Neural Network

Researchers are devoted to integrating syntax information into the NMT system to improve translation performance. Eriguchi et al. [11] used tree LSTM, proposed by Kai et al. [6], to encode the HPSG syntax tree of the source sentence from bottom to top. Chen et al. [14] improved existed encoder with a tree encoder from top to bottom. Chen et al. [12] further extended through a bidirectional tree encoder to learn both sequence and tree structured source representations. Wang et al. [20] proposed a tree-based decoder, simultaneously generates a target-side tree topology and a translation, using the partially-generated tree to guide the translation process. Although these methods have achieved good results, the tree network used by the encoder and decoder makes training and decoding somehow slow and is not suitable for large-scale MT tasks.

There are other works that use syntax information, including grammar concepts, syntax tree structures and dependency units, and syntax trees for attention. Sennrich and Haddow [7] used part-of-speech tags, lemmatized forms and dependency labels to enhance the information carried by each word. In order to better integrate NMT with syntax trees, Eriguchi et al. [8] combined recursive neural network grammar with attention-based NMT system, encouraging models to combine grammatical prior knowledge for translation during training. Li et al. [9] linearized the constituent trees and encoded them with RNN. Wu et al. [10] proposed a sequence-to-dependency NMT model, using two RNNs to jointly generate target translations, and constructing their syntax dependency tree as context to improve word prediction. In order to better integrate NMT with dependency syntax trees, Wu et al. [13] further utilized the global knowledge from the source dependency tree to enrich each encoder state from child to head and head to child. Chen et al. [14] used local dependency unit to extend each source word to capture the long-distance dependency constraints of the source sentence and achieve a good translation of long sentences in NMT. Ahmed et al. [21] design a generalized attention framework for both dependency and constituency trees by encoding variants of decomposable attention inside a Tree-LSTM cell. These methods

used grammar tags to extend source words and provide richer contextual information for word prediction. Due to the linear structure of the RNN, these methods were trained efficiently.

In this paper, we propose a new syntactic branch distance constraint to extend local attention and capture the encoder state associated with the source word syntactically relating to the target word. Rather than improving sequence encoder and decoder with a tree network direclty, we focus on the attention mechanism in the aspect of the syntactic branch distance of syntax tree without making any modifications to specific source representation on the basis of linearized representation using the Tree-LSTM coding syntax tree.

2.2 Attention Mechanism and Local Attention

Neural Machine Translation (NMT) commonly adopts the Encoder-Decoder [1] framework. NMT uses Recurrent Neural Network (RNN) architecture, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) to obtain long-term dependencies. For a given word embedding sequence $X = x_1, x_2, \cdots, x_{|X|}$, encoder typically uses a bidirectional RNN to model the source word sequence and compute a hidden states representation h_i. That is, a forward encoder and a backward encoder encode sequence X to obtain the hidden sequences of the source sentence $H = h_1, h_2, \cdots, h_{|X|}$,

$$\vec{h}_i = f_1\left(\vec{h}_{i-1}, x_i\right) \tag{1}$$

$$\overleftarrow{h}_i = f_2\left(\overleftarrow{h}_{i-1}, x_i\right) \tag{2}$$

$$h_i = \left[\vec{h}_i, \overleftarrow{h}_i\right] \tag{3}$$

where f_1 and f_2 are either GRU(•) or LSTM (•).

The decoder generally adopts conditional RNN with attention mechanism, and predicts the target sentence $Y = y_1, y_2, \cdots, y_{|Y|}$ literally according to the conditional probability $P(y_i)$. The prediction of word in current time step is calculated by the hidden state vector s_t, the last generated word y_{t-1}, and the context vector c_t, using Eq. (5) and (6), where g is a nonlinear function ans f_3 are either GRU(•) or LSTM (•).

And the loss function of NMT model is defined as Eq. (4):

$$\text{loss}_{word} = \sum\nolimits_{t=1}^{Y} - \log p(y_t|y_{<t}; x) \tag{4}$$

$$p(y_t|y_{<t}; x) = g(y_{t-1}, s_t, c_t) \tag{5}$$

$$S_t = f_3(s_{t-1}, y_{t-1}, C_t) \tag{6}$$

The context vector c_t depends on a sequence of source annotations $H = h_1, h_2, \cdots, h_{|X|}$. Each annotation h_i contains information about the whole source word sequence

with a strong focus on the parts surrounding the *i-th* word of the source word sequence. Here we explain below how the context vector c_t are computed in local attention in detail.

Compared with global attention focusing on all context information, local attention selectively focuses on a small context window, which can effectively reduce the computational cost. At the decoding time step i, alignment position p_i is generated for each target word of the batch of sentences using Eq. (7),

$$p_i = S \cdot \text{sigmoid}\left(v^T \tanh(W_p h_i)\right),\ p_i \in [0, S] \tag{7}$$

where S is the length of the source sentence, h_i is the decoder hidden state, and v^T and W_p are model parameters. The context vector c_t is then calculated as the weighted sum of the encoder states within the window $[p_i - D,\ p_i + D]$, where D is the empirical value typically set to 10. Therefore, the weight a_{ij}^l of each source annotation h_i is as follows.

$$\alpha_{ij}^l = \begin{cases} \alpha_{ij} \exp\left(-\frac{(s-p_i)^2}{2\sigma^2}\right), & s \in [p_i - D, p_i + D] \\ 0, & s \notin [p_i - D, p_i + D] \end{cases} \tag{8}$$

The standard deviation σ of the Gaussian distribution is empirically set to $D/2$. In addition, local attention is paid to the source annotations in the window $[p_i - D, p_i + D]$ to calculate the local context vector at current time step. The context vector c_t is then computed as a weighted sum of the annotations h_i:

$$c_i^l = \sum\nolimits_{j \in [p_i - D, p_i + D]} \alpha_{ij}^l h_i \tag{9}$$

It can be seen that the farther away from the center p_i, the lower the weight α_{ij}^l corresponding to source annotation at the position.

3 An Attention Mechanism Based on Syntactic Branch Distance

3.1 Syntactic Branch Distance

Dependency parsing is one of main methods for syntactic analysis. Its basic task is to determine the syntactic structure of a sentence or the interdependence of words in a sentence. Syntactic parsing determines whether the composition of an input sentence conforms to a given grammar, and constructs a syntax tree to represent the structure of the sentence and the relationship between the syntactic components of each level, that is, which words in a sentence constitute a phrase. The dependency syntax tree is a representation of dependency syntax analysis. The dominators and subordinates of dependent syntax tags in the dependent syntax tree are described as parent nodes and child nodes respectively. It expresses formal grammatical rules and constraints as points connected by trees and the information they carry, so that the dependent

syntactic analysis of sentences is transformed into a task of finding a spatially connected structure or a set of dependent pairs of the sentence. In other words, it can well represent a sentence from the perspective of syntactic analysis, and resolve the internal relations among words in the sentence for acquiring sufficient information from the dependency tree. The syntax distance, the connecting distance of any two words in the tree, can be used to describe the close syntax relationship between words. We use Stanford parser[1], which is a Java open source parser based on probabilistic syntax analysis, to acquire dependency pairs between words of a given sentence and generate a dependent syntax tree accordingly.

Generally, the context vector of current time step is obtained by respectively aligning all the encoder states with alignment weights, and the decoder predicts the target word at the next time step by using the context vector. In the traditional attention mechanism, the alignment weight is given by the linear index distance of words in a source sentence. That is to say, in the sequential structure of a sentence, the smaller the index distance between a word and the current source word to be translated, the greater the alignment weight of the word, the greater the contribution it makes to target word prediction when the source word is translated. However, the use of linear index distance is not rigorous, since the linear distance only considers the order in which the words appear in the sentence but ignores the deep structure of the sentence, disregarding the syntactic structure of the sentence and the inter-word dependencies, including composition, context, etc. For example, the three words in Fig. 1, "gave", "went" and "fly" are in the same branch of a dependency tree, and the syntax distances between them are small, <"gave", "went", $d_{syntax} = 1$>, <"gave", "fly", $d_{syntax} = 2$>, <"went", "fly", $d_{syntax} = 1$>. These values indicate that the words have a close syntax relationship, but it is obvious that the linear index distance between them is large, <"gave", "went", $d_{linear} = 5$>, <"gave", "fly", $d_{linear} = 12$>, <"went", "fly ", $d_{linear} = 7$>. Meanwhile, the traditional attention mechanism is inclined to ignore these syntax connections, resulting in translation of the linearly adjacent but less syntax related words are set with greater alignment weights, while words that are more syntax-related and farther away in linear distance are set with less alignment weight, which cause some syntax errors during translation.

To address the mentioned problems, this paper introduces the prior knowledge of syntax tree based on the local attention, and make modifications to the commonly used syntax distances to proposed a new syntactic branch distance, for obtaining more accurate source sentence information when generating target words. Given a source sentence X with dependency tree T, each node represents a source word x_i. For source word as root node, since it has strong syntax relationship with all the words in the sentence, we compute the path length of all remaining words reaching the root word through tree T to obtain syntactic branch distance sequence of source word. That is, this calculates the effective context vector to translate the root node based on the encoder state of all source words and the weighted average of the alignment weights. For source

[1] https://nlp.stanford.edu/nlp.

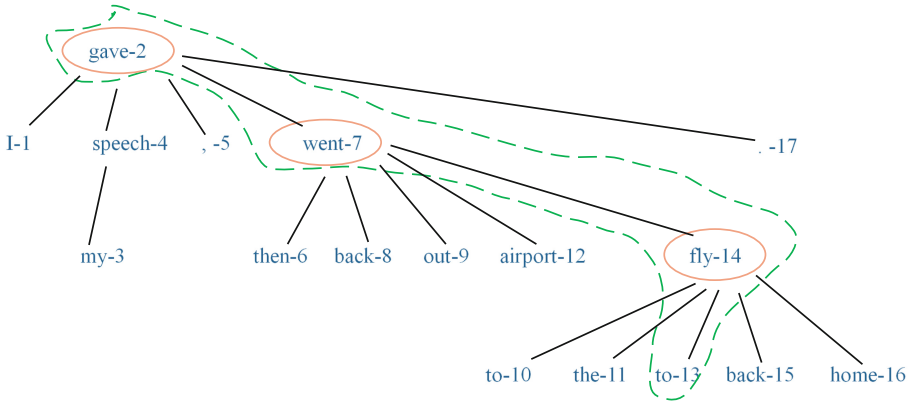

Fig. 1. An example of words in the same branch of a dependency tree

word in leaf node and internal node positions, according to whether the remaining words and current word are in the same branch on the tree T, two situtations are considered. For words that are in the same branch, we define syntactic branch distance of the source word as the level deviations between other words and the source word in syntax tree. For words that are not in the same branch, we set the syntactic branch distance value to the depth of dependency tree of the sentence. The significance of this setting is firstly to reduce the influence of these words on different orders and interference noise to the translated source words. Second, during the translation process, since there still convey many useful information in different branch words, it can be combined with the translated source words to form some phrases, so the empirical value is necessarily set to the depth of syntax tree. Therefore, unwanted noise words can be removed to some extent to ensure proper attention to the word on different branches. Generally, the words on the same branch of a dependency tree are highly correlated with their currently translated source words, thus corresponding alignment weights are large, while the words on other branches have relatively low alignment weights.

As shown in Fig. 2, the syntactic branch distance between the words "*affect*" and "*people*" is 2 for the source word is a root node. For a source word is in leaf node or internal node, the syntactic branch distance between the words "*these*" and "*people*" is 1, while syntactic branch distance between "*these*" and "*dangerous*" is 4 (depth of the dependency tree) for they are not in the same branch. Similarly, each word in the tree is traversed according to the order of source word and the corresponding syntactic branch distance sequence is computed. Finally, all sequences are combined into a syntactic branch distance mask matrix of the sentence. The obtained syntactic branch distance mask matrix is thus shown in Fig. 3.

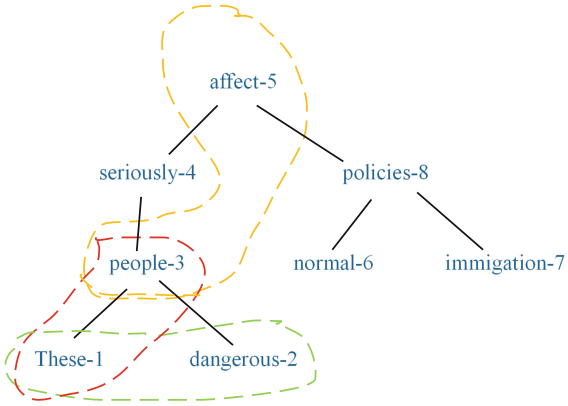

Fig. 2. The dependency syntax tree T and syntactic branch distance calculation for a given stentence (yellow dotted line denotes root nodes, while red and green lines denote the same branch and different branch in leaf and internal nodes, respectively) (Color figure online)

	These	dangerous	people	seriously	affect	normal	immigration	policies
These	0	h	1	2	3	h	h	h
dangerous	h	0	1	2	3	h	h	h
people	1	1	0	1	2	h	h	h
seriously	h	2	1	0	1	h	h	h
affect	3	3	2	1	0	2	2	1
normal	h	h	h	h	2	0	h	1
immigration	h	h	h	h	2	h	0	1
policies	h	h	h	h	1	1	1	0

Fig. 3. The syntactic branch distance mask matrix M of the sentence (Each line represents a syntactic branch distance mask for a source word, where h is the depth of syntax tree)

3.2 The Attention Mechanism Based on Tuned Branch Syntax Distance

In order to solve the problem of inaccurate focused source context of local attention, we propose an attention mechanism based on a new syntactic branch distance, aiming at integrating accurate and effective syntax knowledge with attention mechanism to improve the accuracy of source-side context information.

We use the seq2seq model framework, which mainly consists of a encoder model by a bidirectional RNN, a decoder model by a conditional RNN and a generator which

depend on conditional probability. Besides, further improvement work is conducted in the attention mechanism. First, the alignment source position p_i is learned for each target word by the Eq. (7) at the current decoding time step i. After that, the alignment weights constrain by syntactic branch distance through source position p_i and syntactic branch distance matrix M are calculated using Eq. (10):

$$e_{ij}^{bs} = e_{ij} \exp\left(-\frac{\left(M_{[p_i][j]}\right)2}{2\sigma^2}\right) \tag{10}$$

Furthermore, the standard deviation σ is set to $h/2$ in our experiments, where h is empirically set to be the depth of syntax tree of a given sentence and it is similar to the order of syntax tree level. First of all, the syntactic branch distance value of a word not in the same branch is set to be depth of dependency tree of the sentence. Secondly, all syntactic branch distances of words can be obtained from the hierarchy of a syntax tree. In order to remove unwanted noise words to some extent without losing proper attention to words on different branches, we set the largest syntactic branch distance to be the depth number of syntax tree.

l is the length of the sentence, and $\alpha_{ij}^{bs_n}$ is normalized considering all the syntactic branch distances of current source word, i.e., the row of syntactic branch distance mask corresponding to the current source word.

$$\alpha_{ij}^{bs_n} = \frac{\exp\left(e_{ij}^{bs}\right)}{\sum_{k \in M_{[p_i][k]} < h} \exp\left(e_{ik}^{bs}\right)}, \; j \in [0, l] \tag{11}$$

Finally, the context vector c_i^{bs} is calculated as the weighted sum of the source annotations of attention by the weights alignment of the attention of single grammar branch distance.

$$c_i^{bs} = \sum_{j}^{J} \alpha_{ij}^{bs_n} h_j \tag{12}$$

4 Experiments and Results

4.1 Expreiment Settings

To evaluate the effectiveness of our proposed model, the commonly applied standard dataset IWSLT 2017[2] is used as the evaluation dataset. 204936 dual-language sentences in English and German is used as training data. The supplementary dev2010 dataset is as the validation data set, and tst2010, tst2011, tst2012, tst2013, tst2014 are used as testing data sets.

[2] https://sites.google.com/site/iwsltevaluation2017/Dialogues-task.

We use a local attention proposed by Luong et al. [5] as a baseline method. The local attention is improved on the basis of global attention. The local attention mechanism selectively focuses on the context of a window in which current source word is located in, and considers that the context can benefit decoder on the prediction of next generated word. Experiments prove that it not only reduce the computational cost, but also outperforms global attention on translation performance in terms of BLEU score.

The NMT model used in the experiment is implemented based on Nematus codes by Sennrich et al. [16]. We use the Stanford parser (Chang et al. [17]) to generate dependency trees for source sentences. Our model limits the source and target vocabulary size to 50 K and the maximum training sentence length to 50. We randomly shuffle our training data set in each epoch. The batch size is 40, the word embedding dimension is 512-dimensions, the hidden layer dimension is 1024-dimensions, and the decoded beam size is 12. The default dropout technique in Nematus is used on all the layers (Hinton et al. [18]). Our NMT model choose ADADELTA as the optimizer (Zeiler et al. [15]), and trains about 400,000 small batches. It runs on a single GeForce GTX 1080 GPU for 2 days. The case-sensitive 4-gram NIST BLEU score (Papineni et al. [19]) is used as the evaluation metric.

4.2 The Results

The performance comparison of our model with the baseline is conducted and the results is shown as Table 1. From the table, the translation results of attention NMT based on the syntactic branch distance constraint (as SbdAtt) on the IWSLT 2017 testing dataset is 23.49. Compared with global attention (as GlobalAtt), our proposed LocalAtt-SBD has increased 1.61 BLEU points on average. This indicates that, compared with global attention focusing on global information, our method acquires more accurate context information during the translation process, which effectively improves translation performance.

Table 1. Results on EN-DE translation tasks of different attention mechanism

EN-DE	dev2010	tst2010	tst2011	tst2012	tst2013	tst2014	tst2015	avg
GlobalAtt	19.87	21.94	24.45	21.93	22.72	20.05	22.22	21.88
LocalAtt	20.31	21.05	22.56	20.69	22.11	19.36	21.22	21.04
SbdAtt	**22.67**	**24.00**	**25.29**	**22.54**	**25.02**	**21.42**	**23.55**	**23.49**

In terms of the baseline local attention (as LocalAtt), our proposed LocalAtt-BSD has increased by 2.45 BLEU points on average, demonstrating that our method can learn more source dependency information to effectively improve the translation performance of NMT. The proposed syntactic branch distance attention can capture more translation information than linear distance attention to improve word prediction.

5 Conclusion

This paper tried to integrate the prior knowledge of syntactic analysis with traditional attention mechanism to improve translation performance. An attention mechanism based on tuned branch syntax was proposed. Syntax-directed selective attention on the word associated with source word, including the cases of the same branch and different branch with the source word, was proposed for the predication of target words. Experiment results on the IWSLT2017 showed that the proposed model outperformed the local attention baseline method. In the future, we will extend the experiment to other languages (such as Chinese-English) to test the scalability of the model and the applicability on long sentences.

Acknowledgements. This work was supported by National Natural Science Foundation of China (No.61772146).

References

1. Kalchbrenner, N., Blunsom, P.: Recurrent continuous translation models. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1700–1709 (2013)
2. Cho, K., Merrienboer, B.V., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
3. Sutskever, I., Vinyals, O., Le, Q.V.: Sutskever, I., et al.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2015)
5. Luong, M.T., Sutskever, I., Le, Q.V., et al.: Addressing the rare word problem in neural machine translation. Bull. Univ. Agric. Sci. Vet. Med. Cluj-Napoca. Vet. Med. **27**(2), 82–86 (2014)
6. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075 (2015)
7. Sennrich, R., Haddow, B.: Linguistic input features improve neural machine translation. In: Proceedings of the First Conference on Machine Translation, Berlin, Germany, pp. 83–91. ACL (2016)
8. Eriguchi, A., Tsuruoka, Y., Cho, K.: Learning to parse and translate improves neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, pp. 72–78. ACL (2017)
9. Li, J., Xiong, D., Tu, Z., Zhu, M., Zhou, G.: Modeling source syntax for neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, pp. 688–697. ACL (2017)
10. Wu, S., Zhang, D., Yang, N., Li, M., Zhou, M.: Sequence-to-dependency neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1), Vancouver, Canada, pp. 698–707 (2017)

11. Eriguchi, A., Hashimoto, K., Tsuruoka, Y.: Tree-to-sequence attentional neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, pp. 823–833 (2016)
12. Chen, H., Huang, S., Chiang, D., Chen, J.: Improved neural machine translation with a syntax-aware encoder and decoder. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, pp. 1936–1945 (2017)
13. Wu, S., Zhou, M., Zhang, D.: Improved neural machine translation with source syntax. In: Proceedings of the Twenty Sixth International Joint Conference on Artificial Intelligence, IJCAI-2017, pp. 4179–4185 (2017)
14. Chen, K., Wang, R., Utiyama, M., Liu, L., Zhao, T., et al.: Neural machine translation with source dependency representation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, pp. 23–32 (2017)
15. Zeiler M.D.: ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
16. Sennrich, R., Firat, O., Cho, K., et al.: Nematus: a toolkit for neural machine translation. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, pp. 65–68 (2017)
17. Chang, P.C., Tseng, H., Jurafsky, D., Manning, C.D.: Discriminative reordering with Chinese grammatical relations features. In: Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation, Boulder, Colorado, pp. 51–59 (2009)
18. Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
19. Papineni, K., Roukos, S., Ward, T., Zhu, W.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp. 311–318 (2002)
20. Wang, X., Pham, H., Yin, P., Neubig, G.: A tree-based decoder for neural machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4772–4777(2018)
21. Ahmed, M., Samee, M. R., Mercer, R. E.: Improving tree-LSTM with tree attention. In: Proceedings of the 2019 IEEE 13th International Conference on Semantic Computing, pp. 247–254(2019)