Research Article

Ping Wang*

A study of an intelligent algorithm combining semantic environments for the translation of complex English sentences

https://doi.org/10.1515/jisys-2022-0048
received December 19, 2021; accepted March 30, 2022

Abstract: In order to improve the translation quality of complex English sentences, this paper investigated unknown words. First, two baseline models, the recurrent neural machine translation (RNMT) model and the transformer model, were briefly introduced. Then, the unknown words were identified and replaced based on WordNet and the semantic environment and input to the neural machine translation (NMT) model for translation. Finally, experiments were conducted on several National Institute of Standards and Technology (NIST) datasets. It was found that the transformer model significantly outperformed the RNMT model, its average bilingual evaluation understudy (BLEU) value was 42.14, which was 6.96 higher than the RNMT model, and its translation error rate (TER) was also smaller. After combining the intelligent algorithm, the BLEU values of both models improved, and the TER became smaller; the average BLEU value of the transformer model combined with the intelligent algorithm was 43.7, and the average TER was 57.68. The experiment verifies that the transformer model combined with the intelligent algorithm is reliable in translating complex sentences and can obtain higher-quality translation results.

Keywords: semantic environment, English translation, complex sentence, transformer model, bilingual evaluation understudy

1 Introduction

Natural language processing (NLP) is an important element of artificial intelligence [1], including text classification, machine translation (MT), etc. [2]. As the world becomes more and more globalized and internationalized, academic and cultural exchanges between different countries are more frequent; therefore, the demand for translation is further expanded, and MT has also been developed rapidly [3]. MT is a method to achieve conversion between different languages through computer technology. However, it always has the problem of a low accuracy rate, which is difficult to meet the needs of real life, especially in some professional fields, such as the translation of scientific and technical literature. Therefore, how to improve the quality of MT has become the focus of researchers. Xiong et al. [4] designed a topic-based coherence model that automatically extracted coherent chains for every text to be translated, integrated the model into the phrase-based MT model, and found through extensive experiments that the method yielded more coherent results that were more similar to the reference translation. Bicici and Yuret [5] used a feature decay algorithm to optimize MT and found that the method had a high bilingual evaluation understudy (BLEU) value and reduced the computing time to about half after performing experiments on two million English–German sentence pairs from the Europarl corpus. Wu et al. [6] proposed a grammar-aware encoder

* **Corresponding author: Ping Wang,** School of Foreign Language, Zhengzhou Tourism College, No. 188, Jinlong Road, Zhengdong New District, Zhengzhou, Henan 450000, China, e-mail: p0r9wa@163.com

to merge source dependency trees into neural machine translation (NMT) and found that the method improved the translation quality through experiments on several translation tasks. Hewavitharana and Vogel [7] designed a phrase alignment method that bypassed the non-parallel part of sentences and aligned only the parallel part. They found an improvement of 1.2 BLEU over the baseline by translating Arabic–English and Urdu–English. Yu et al. [8] put forward a regularization method for back-translation. To enhance the robustness of sentences after autoencoding or back-translation, they applied the adversarial attack on representations. The experiment found that the method outperformed the cross-lingual language model baseline by 0.4–1.8 BLEU scores. Sun and Yong [9] studied the Tibetan–Chinese MT. Through studies such as data augmentation and attention mechanism and taking seq2seq and Transformer models as the baseline, the BLEU value of Tibetan–Chinese NMT was increased from 5.53 to 19.03. With its advantages in terms of efficiency and quality, NMT has become one of the most mainstream translation methods at present [10]; however, there are still some problems with conveying the semantic information completely. Semantics is always a very complex problem in translation [11]. This paper designed an intelligent algorithm for unknown word processing based on the semantic environment and combined it with the NMT model to translate complex English sentences. The experiment verified the effectiveness of the NMT model combined with the intelligent algorithm. This work makes some contributions to further improving the quality of NMT.

2 Intelligent algorithms that combine semantic environments

2.1 Baseline model

In this paper, two mainstream NMT models were used as baseline models for translating complex English sentences. One was a recurrent neural machine translation (RNMT) model, and the other was a transformer model based on a self-attentive mechanism. These two models are briefly described next.

The RNMT model [12] directly translated source sentences into target sentences. Before a source-side sentence entered the encoder, it was written as a word vector sequence:

$$X = \{x_1, x_2, \ldots, x_n\}, \tag{1}$$

where X is the source sentence, n is the total number of words in the sentence, and x_i refers to the word vector of the i-th word. The encoder of the RNMT model was a bidirectional RNN, and the forward RNN reads the source sentence from left to right and outputs the forward hidden state:

$$\overrightarrow{h} = \{\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots, \overrightarrow{h_n}\}. \tag{2}$$

The calculation formula of $\overrightarrow{h_i}$ is

$$\overrightarrow{h_i} = f(\overrightarrow{h_{i-1}}, x_i), \tag{3}$$

where f is the nonlinear activation function.

Similarly, the reverse RNN reads the source sentence from right to left and output the reverse hidden state:

$$\overleftarrow{h} = \{\overleftarrow{h_1}, \overleftarrow{h_2}, \ldots, \overleftarrow{h_n}\}. \tag{4}$$

The hidden state of the source statement was obtained after splicing the above two states:

$$h = \{h_1, h_2, \ldots, h_n\}, \tag{5}$$

$$h_i = [\overrightarrow{h_i} \cdot \overleftarrow{h_i}]. \tag{6}$$

Then, the decoder used a one-way RNN to predict the target sentence: $\{y_1, y_2, \ldots, y_m\}$. The calculation formula of the i-th predicted word y_i is

$$p(y_i|y_1, y_2, \ldots, y_{i-1}, x) = g(y_{i-1}, s_i, c_i), \tag{7}$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i), \tag{8}$$

where s_i refers to the current hidden state and c_i refers to the current source context vector. The calculation formula of c_i is

$$c_i = \sum_j a_{ij} h_j, \tag{9}$$

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})}, \tag{10}$$

$$e_{ij} = a(s_{i-1}, h_j), \tag{11}$$

where a_{ij} is the attention weight, h_j is the hidden state of the j-th word of the source sentence, and e_{ij} is the matching degree between the i-th output word and the j-th input word.

The transformer model [13] also consisted of two parts, and it differed most from the RNMT model in that it used a multi-head attention mechanism [14] that represented every word by three vectors: Query (Q), Key (K), and Value (V). A multi-head attention mechanism contained multiple dot-product attentions. The calculation of the dot-product attention is

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{12}$$

where d_k refers to the dimensionality of K. Then, multiple dot-product attentions are spliced and put into a feedforward neural network (FFN):

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{13}$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_i)W^o, \tag{14}$$

If there were training parameters, W_1, W_2, b_1, and b_2, FFN took rectified linear unit as the activation function, then

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2. \tag{15}$$

Finally, the transformer model used the positional encoding method to add location information, and the calculation formula is

$$\text{PE}(\text{pos}, 2i) = \sin(\text{pos}/10{,}000^{2i/d_{\text{model}}}), \tag{16}$$

$$\text{PE}(\text{pos}, 2i + 1) = \cos(\text{pos}/10{,}000^{2i/d_{\text{model}}}), \tag{17}$$

where pos refers to the position of a word in a sentence and i is the vector dimension.

2.2 Translation algorithms that combine semantic environments

In MT, there are some common problems, such as unknown words, missed translations, and over-translations [15], which leads to poor translation quality. This paper mainly analyzed the problem of unknown words and designed an intelligent algorithm by combining the semantic environment.

Unknown words refer to words not covered by the word list. Models cannot correctly interpret unknown words in the translation process and uniformly replace them with "<UNK>," which is prone to ambiguity and affects the translation results. This paper translated unknown words by replacing them with suitable words obtained through calculating semantic similarity based on the semantic environment. Based on the

WordNet semantic network [16], the synonym of unknown words was found from WordNet after processing the source sentence by word segmentation. In the WordNet semantic environment, there are a variety of semantic concept relationships, and two relationships were involved in this paper.

(1) Synonymy: words that are identical or similar on the level of meaning.
(2) Hypernymy/hyponymy: if a word is a subclass of another word, there is a hypernymy/hyponymy relationship between them. Taking "car" and "vehicle" as examples, "car" is the hyponym of "vehicle," and "vehicle" is the hypernym of "car."

The specific process of the intelligent algorithm combined with the semantic environment is as follows.

(1) For a word w whose part of speech was w_{pos}, all words whose part of speech was w_{pos} were screened out from WordNet and denoted as synsets_w. Then, known words were screened out from those words and added to the synonym set of w.
(2) If the synonym set was empty, all the hypernyms of synsets_w were found out and denoted as hyber_synsets_w. Then, all the synonyms of the hypernym set were found out, and the known words were added to the synonym set of w.
(3) If the synonym set was empty, all the hyponyms of synsets_w were found out and denoted as hypo_synsets_w. Then, all the synonyms of the hyponym set were found, and the known words were added to the synonym set of w.
(4) According to the above steps, the synonym set of w was found as the candidate replacement word set w_i'. The calculation of semantic similarity was based on the ternary language model, and the calculation formula is

$$\text{Score}_{3\text{-gram}}(w_i', w_i) = \frac{p(w_i'|w_{i-1}w_{i-2}) + p(w_{i+1}|w_i'w_{i-1}) + p(w_{i+2}|w_{i+1}w_i')}{3}. \tag{18}$$

The similarity score of two words was calculated based on their distance in WordNet, and the formula is

$$\text{Score}_{\text{WordNet}}(w_i', w_i) = \frac{1}{\text{path}(w_i', w_i) + 1}, \tag{19}$$

where $\text{path}(w_i', w_i)$ refers to the distance between two words. Finally, the calculation formula of semantic similarity of (w_i', w_i) is

$$\text{sim}(w_i', w_i) = \text{Score}_{n-\text{gram}}(w_i', w_i) \cdot \text{Score}_{\text{WordNet}}(w_i', w_i). \tag{20}$$

Then, the replacement word for the unknown word is

$$w_{\text{best}} = \underset{w' \int S}{\text{argmax}}\, \text{sim}(w_i', w_i). \tag{21}$$

(5) The replaced sentence was input into the NMT model for translation. After the result was obtained, the original unknown word was put back. In this paper, a word alignment model was trained by GIZA++. A bilingual dictionary that includes all the known words was selected. For word t_i in the translation result, if its aligned word s_j was obtained by replacement and the original unknown word was s_j', then the alignment relationship was determined by the bilingual dictionary. If (s_j, t_i) was in the dictionary, then the alignment relationship was proved to be correct, and t_i was replaced by the translation of s_j'.

3 Experimental analysis

3.1 Experimental setup

The Chinese Treebank 6.0 corpus was used, including 1,067 files, 20,367 sentences, 647,523 English words, and 963,461 Chinese words. The development set for the experiment was National Institute of Standards

and Technology (NIST) 05 in the Linguistic Data Consortium dataset. The test sets were NIST 02, 03, 04, 06, and 08. The specific data are shown in Table 1.

Table 1: Experimental data

	Chinese sentences	English sentences
NIST 05	1,082	4,328
NIST 02	878	3,512
NIST 03	919	3,676
NIST 04	1,788	7,152
NIST 06	1,664	6,656
NIST 08	1,357	5,428

The word alignment was implemented with GIZA++. The RNMT model was RNNSearch[7] [17]; the decay constant was 0.95; the denominator constant was 10^{-6}; the word vector dimension was 620; the hidden dimension was 1,000; the output layer used the dropout strategy; the dropout rate was 0.5; and the beam_size was 10 at decoding. The transformer model was the tensor2tensor model [18], with six encoders, six decoders, and eight multi-head attention mechanisms; the word vector dimension was 512; the hidden dimension was 512; the internal dimension of FFN was 2,048; the beam_size was 4 at decoding; and the length penalty was 0.6.

3.2 Evaluation indicator

The indicators used were BLEU and translation error rate (TER).

(1) BLEU: its principle was comparing the reference translation with the MT. The value of BLEU was between 0 and 1; the larger the value was, the higher the translation quality was. First, the matching accuracy p_n of the n-gram was calculated:

$$p_n = \frac{\sum_{c\in\{\text{candidates}\}}\sum_{N\text{-gram}\in c}\text{count}_{\text{clip}}(N\text{-gram})}{\sum_{c'\in\{\text{candidates}\}}\sum_{N\text{-gram}\in c'}\text{count}_{\text{clip}}(N\text{-gram}')}, \tag{22}$$

where $\text{count}_{\text{clip}}(N\text{-gram})$ refers to the number of occurrences of n-gram in the reference translation.

There was also a length penalty term BP in the BLEU value:

$$\text{BP} = \begin{cases} 1, c > r, \\ e^{1-r/c}, c \le r, \end{cases} \tag{23}$$

where c is the length of the reference translation and r is the length of the MT.

Ultimately, the formula for calculating the BLEU value is

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{24}$$

where N is the maximum order of n-gram grammar and w_n is the weight of co-occurring n-gram, generally $1/n$. In MT evaluation, 4-tuple is usually used; therefore, $N = 4$ in this paper.

(2) TER: it is a common indicator used for evaluating the performance of MT. Its principle was evaluating the number of different words in MT and reference translation. The smaller the TER was, the smaller the error rate of MT was, i.e. the higher the translation quality was.

3.3 Experimental results

The BLEU values of the baseline models and the models combined with the intelligent algorithm on different datasets were compared, as shown in Table 2.

Table 2: Comparison results of BELU values on different datasets

	NIST 05	NIST 02	NIST 03	NIST 04	NIST 06	NIST 08
The RNMT model	35.12	37.67	36.59	38.95	35.87	26.89
The transformer model	43.68	42.12	43.25	45.67	42.27	35.84
The intelligent algorithm + RNMT model	39.33	39.56	38.46	40.88	37.64	29.12
The intelligent algorithm + Transformer model	44.61	44.89	44.16	47.62	44.12	36.79

It is seen from Table 2 that before combining the intelligent algorithm, there was a gap in the BLEU value between the two baseline models; on the development set, the BLEU values of the RNMT model and the transformer model were 35.12 and 43.68, respectively, i.e. the transformer model was 8.56 larger than the RNMT model, and the situation was the same on the test set, indicating that the transformer model had better MT performance than the RNMT model. After combining the intelligent algorithm, the BLEU values of both algorithms showed an improvement; for example, in the development set, the BLEU value of the intelligent algorithm + RNMT model was 39.33, which showed an improvement of 4.21, and the BLEU value of the intelligent algorithm + transformer model was 44.61, which showed an improvement of 0.93; the situations were the same on the test set, indicating that the intelligent algorithm designed in this paper was effective in improving the translation quality. In addition, since the transformer model has performed better than the RNMT model before improvement, its improvement was not significant as the RNMT model.

The average BLEU values of different algorithms were calculated, and the results are shown in Figure 1.

It was seen from Figure 1 that the average BLEU value of the RNMT model and the transformer model was 35.18 and 42.14, i.e. the transformer model was 6.96 larger than the RNMT model. After combining the intelligent algorithm, the average BLEU value of the intelligent algorithm + RNMT model reached 37.5, which showed an increase of 2.32 compared to the RNMT model, while the average BLEU value of the intelligent algorithm + transformer model was 43.7, which showed an increase of 1.56 compared to the transformer model and an increase of 6.2 compared to the intelligent algorithm + RNMT model. The experimental results demonstrated the advantages of the transformer model for MT and the reliability of the intelligent algorithm combined with the semantic environment designed in this paper for improving translation quality.

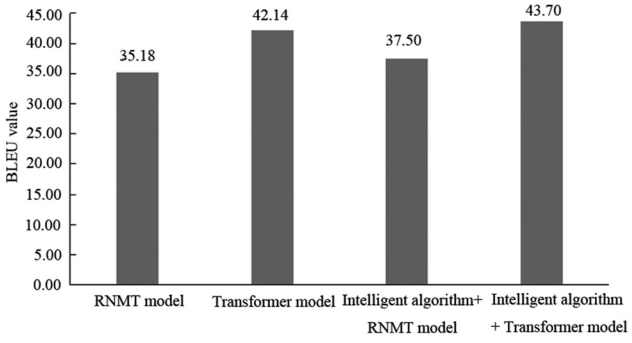

Figure 1: Comparison results of average BLEU values between different algorithms.

The average TER was compared between different algorithms, and the results are shown in Figure 2.

It is seen from Figure 2 that the average TER of the transformer model was 1.39 lower than that of the RNMT model (59.33 vs 60.72); the average TER of the intelligent algorithm + RNMT model was 59.37, which

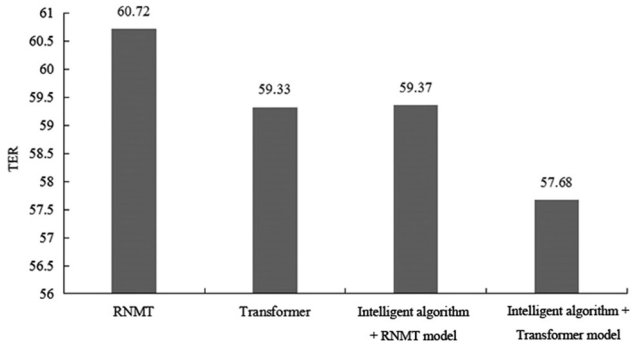

Figure 2: The comparison of the average TER between different algorithms.

was 1.35 lower than that of the RNMT model; the average TER of the intelligent algorithm + transformer model was 57.68, which was 1.65 lower than that of the transformer model and 1.69 lower than that of the intelligent algorithm + RNMT model. The above results demonstrated that the MT obtained by the intelligent algorithm + RNMT model had fewer mistakes and higher quality.

Two complex statements were used as examples to analyze the details of the translation, as shown in Table 3.

Table 3: Example sentence analysis

Source sentence	While a large number of birds have died for unknown causes in the past week in the southern United States, Swedish officials said as many as 100 birds had fallen today on a snow-covered(unk) street in Sweden as well
Reference translation	在美国南部1周来有大批飞鸟不明死亡的同时，瑞典官员说，瑞典1条白雪皑皑的街道上，今天也坠落了至多100只飞鸟。
RNMT	过去一周，美国南部有大量鸟类因不明原因死亡，瑞典官员说，今天也有多达100只鸟掉在瑞典的UNK街道上。
Transformer	过去一周，美国南部有大量鸟类因不明原因死亡，瑞典官员说，今天瑞典的一条UNK街道上也有多达 100 只鸟掉下来。
Intelligent algorithm + RNMT	过去一周，美国南部有大量鸟类因不明原因死亡，瑞典官员说，今天有多达100只鸟降落在瑞典被雪覆盖的街道上。
Intelligent algorithm + transformer	过去一周，美国南部有大量鸟类因不明原因死亡，瑞典官员说，今天有多达100只鸟坠落在瑞典白雪皑皑的街道上。
Source sentence	Manila removed an intelligence officer from the police, for he had released unconfirmed intelligence(unk) about the terrorist threat to the Australia and Canadian embassies.
Reference translation	马尼拉曾免除了警方情报官员的职务，因为他透露有关澳洲与加拿大大使馆遭到恐怖威胁的未经证实的情报。
RNMT	马尼拉将一名情报官员从警方中带走，因为他公布了有关澳大利亚和加拿大大使馆恐怖威胁的未经证实的UNK。
Transformer	马尼拉从警察局撤掉了一名情报官员，因为他向澳大利亚和加拿大大使馆泄露了未经证实的恐怖主义威胁UNK。
Intelligent algorithm + RNMT	马尼拉解除了一名情报官员的职务，因为他向澳大利亚和加拿大大使馆发布了有关恐怖威胁的未经证实的情报。
Intelligent algorithm + transformer	马尼拉将一名情报官员从警方中撤职，因为他向澳大利亚和加拿大大使馆发布了有关恐怖主义威胁的未经证实的情报。

In Table 3, the first example sentence contained an unknown word, “snow-covered,” which was not translated in the translation of the baseline system but was correctly translated after combining with the intelligent algorithm. In addition, the translation of the transformation model was more fluent than that of the RNMT model; for example, the translation of the word “fallen” was more contextual in the transformer

model. In the second example sentence, which also contained an unknown word, “intelligence,” RNMT and transformer models did not translate it. In addition, the RNMT model translated “removed” to “带走,” which was not appropriate enough. In terms of word order and semantics, the results of the intelligent algorithm combined model were closer to the reference translation, which verified the reliability of the model in translating complex English long sentences.

4 Conclusion

This paper designed an intelligent algorithm combining the semantic environment for the problem of unknown words and combined it with the NMT model to translate complex English sentences. The comparison experiments on different datasets showed that the transformer model exhibited better performance in translation. After combining the intelligent algorithm, the BLEU values of both models significantly improved, and the TER decreased. It was also found from the detailed analysis of the translation results that the NMT model combined with the intelligent algorithm got more fluent translation results, which verified the effectiveness of the model. The NMT model combined with the intelligent algorithm can be further promoted and applied in practice. However, this paper has some limitations, such as only the processing of unknown words was studied and only word-level replacement was considered in unknown word replacement. In future research, issues such as low-frequency words can be studied, and in addition, phrase-level replacement methods can be considered to improve the NMT model further so that it can perform better in the translation of complex English sentences. The research in this paper provides some theoretical bases for further research on the NMT model, which can promote researchers to further research semantic issues in intelligent translation.

Conflict of interest: Author states no conflict of interest.

References

[1] Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing. IEEE Comput Intell M. 2018;13:55–75.
[2] Ghareb AS, Bakar AA, Hamdan AR. Hybrid feature selection based on enhanced genetic algorithm for text categorization. Expert Syst Appl. 2016;49:31–47.
[3] Peris A, Domingo M, Casacuberta F. Interactive neural machine translation. Comput Speech Lang. 2016;45:201–20.
[4] Xiong D, Zhang M, Wang X. Topic-based coherence modeling for statistical machine translation. IEEE/ACM T Audio Spe. 2015;23:483–93.
[5] Bicici E, Yuret D. Optimizing instance selection for statistical machine translation with feature decay algorithms. IEEE/ACM T Audio Spe. 2015;23:339–50.
[6] Wu S, Zhang D, Zhang Z, Yang N, Li M, Zhou M. Dependency-to-dependency neural machine translation. IEEE/ACM T Audio Spe. 2018;26:2132–41.
[7] Hewavitharana S, Vogel S. Extracting parallel phrases from comparable data for machine translation. Nat Lang Eng. 2016;22:549–73.
[8] Yu H, Luo H, Yi Y, Cheng F. A2R2: robust unsupervised neural machine translation with adversarial attack and regularization on representations. IEEE Access. 2021;9:19990–8.
[9] Sun Y, Yong C. Research on Tibetan–Chinese neural network machine translation with few samples. J Phys Conf Ser. 2021;1871:012095(8pp).
[10] Lee J, Cho K, Hofmann T. Fully character-level neural machine translation without explicit segmentation. Trans Assoc Comput Ling. 2017;5:365–78.
[11] Chua CC, Lim TY, Soon LK, Tang EK, Ranaivo-Malançon B. Meaning preservation in example-based machine translation with structural semantics. Expert Syst Appl. 2017;78:242–58.
[12] Qiao Y, Hashimoto K, Eriguchi A, Wang H, Wang D, Tsuruoka Y, et al. Parallelizing and optimizing neural Encoder–Decoder models without padding on multi-core architecture. Future Gener Comp Sy. 2018;1206–13.

[13] Liu HI, Chen WL. Re-transformer: a self-attention based model for machine. Trans Proc Comput Sci. 2021;189:3–10.
[14] Lin F, Zhang C, Liu S, Ma H. A hierarchical structured multi-head attention network for multi-turn response generation. IEEE Access. 2020;8:46802–10.
[15] Sulaeman MA, Purwarianti A. Development of Indonesian-Japanese statistical machine translation using lemma translation and additional post-process. International Conference on Electrical Engineering & Informatics; 2015. p. 54–8.
[16] Wei T, Lu Y, Chang H, Zhou Q, Bao X. A semantic approach for text clustering using WordNet and lexical chains. Expert Syst Appl. 2015;42:2264–75.
[17] Bastien F, Lamblin P, Pascanu R, Bergstra J, Goodfellow I, Bergeron A, et al. Theano: new features and speed improvements; 2012. doi: 10.48550/arXiv.1211.5590.
[18] Vaswani A, Bengio S, Brevdo E, Chollet F, Gomez A, Gouws S, et al. Tensor2Tensor for neural machine translation; 2018. doi: 10.48550/arXiv.1803.07416.