In recent years, with the development of deep learning, machine translation using neural network has gradually become the mainstream method in industry and academia. The existing Chinese-English machine translation models generally adopt the deep neural network architecture based on attention mechanism. However, it is still a challenging problem to model short and long sequences simultaneously. Therefore, a bidirectional LSTM model integrating attention mechanism is proposed. Firstly, by using the word vector as the input data of the translation model, the linguistic symbols used in the translation process are mathematized. Secondly, two attention mechanisms are designed: local attention mechanism and global attention mechanism. The local attention mechanism is mainly used to learn which words or phrases in the input sequence are more important for modeling, while the global attention mechanism is used to learn which layer of expression vector in the input sequence is more critical. Bidirectional LSTM can better fuse the feature information in the input sequence, while bidirectional LSTM with attention mechanism can simultaneously model short and long sequences. The experimental results show that compared with many existing translation models, the bidirectional LSTM model with attention mechanism can effectively improve the quality of machine translation.

1. Introduction

Machine translation is an important part of natural language processing. Machine translation (MT) mainly studies the process of how to automatically translate one human language into another by computer [1–5], so as to realize the mutual conversion between different languages. With the rapid development of science and technology and social economy, the cultural exchanges at home and abroad are increasing, which leads to the increasing demand for Chinese-English translation. Google, Microsoft, Baidu, Sogou, and other companies are constantly developing and perfecting machine translation systems. Compared with human translation, machine translation, as a low-cost and efficient communication method, has become an indispensable part of the translation industry. The mainstream machine translation models mainly include statistical machine translation (SMT) [6–8] and neural machine translation (NMT) [9–11]. The traditional machine translation method is the SMT model, which counts information such as word pairs, parallel phrase pairs, and parallel syntactic structures from large-scale parallel corpus to establish a statistical model for the translation process. The research methods mainly include word-based statistical methods [12], phrase-based statistical methods [13], and syntactic structure-based statistical methods [14]. Among them, the phrase-based statistical machine translation takes phrases (that is, any consecutive words) as the basic translation unit, which can well solve the dependency relationship between the local contexts of sentences, and the translation quality has been greatly improved compared with the word-based statistical method. In recent years, with the development of deep learning, a translation model using neural network to map Chinese to English has emerged, which is called NMT...
model [15, 16]. NMT model significantly improves the quality of machine translation, surpasses the performance of traditional SMT methods, and becomes the mainstream method in industry and academia at present. Different from the traditional SMT, NMT is committed to building a separate neural network to achieve the best translation performance through joint training and adjustment. The benchmark NMT system is the encoder-decoder framework [17], which uses bilingual parallel corpus to realize the end-to-end [18] training process. Coding and decoding functions are realized by recurrent neural networks (RNN) [19, 20]. These two cyclic neural networks are modeled by an attention layer connection, which detects the information related to all the words at the source when translating the target words. This process is called attention mechanism [21]. RNN is a connection model, which can capture the dynamic information of input sequence through cyclic operation in network nodes. Unlike the standard feedforward neural network, the circular neural network can maintain a state, which can represent information in any long context window. Gulcehre et al. [22] proposed the method of integrating RNN language model into NMT. Sano et al. [23] expanded the RNN structure, transferred the derivative to historical information, and improved the memory ability of translation model for long-distance information. In view of the difficulty in introducing translation rules into NMT, Wu et al. [24] coded the rules and selected the rules through attention mechanism in the process of translation, which achieved good translation results, but it also caused high time complexity. In recent years, the long short-term memory (LSTM) network has made breakthrough progress in many learning tasks in the computer field, such as adding subtitles to images, language translation, and handwriting font recognition. Ren et al. [25] extended the LSTM network structure to 16 layers, which made the single model translation result better than SMT. However, the general neural network model will not save the intermediate information and will give the same weight to the short sequence or phrase in the input sequence. Therefore, this paper proposes an attention bidirectional LSTM (A-BLSTM) model which integrates attention mechanism. Because different words and phrases in an input sequence will contain different amounts of information, the A-BLSTM model contains a local attention mechanism. Local attention mechanism is mainly used to learn which word or phrase in the input sequence contains more information. Global attention mechanism is used to learn which layer of representation vector of input sequence should be given more attention (weight). Our main contributions are as follows: Firstly, for the simultaneous modeling of short sequences and long sequences, a hierarchical structure integrating attention mechanism is proposed, which can obtain multilayer intermediate representation vectors of input sequences instead of a single fixed-length word vector. Secondly, the number of network layers in A-BLSTM model can change with the complexity of tasks. Finally, the local attention mechanism can effectively select relatively important words or phrases, and the global attention mechanism can effectively select more credible intermediate representation vectors.

2. Word Vector Generation

To transform the machine translation problem into a machine learning problem, it is first necessary to mathematize the language symbols used in the translation process. As the input data of the translation model, word vector has a great influence on the quality of the final model. As the transmission form of data in NMT, the acquisition of word vector is the basis of all research work.

2.1. Word Vector Model. Skip-gram model is used for word vector model. Skip-gram model [26], also known as skip model, takes a word as input and predicts its context around a text sequence. For example, for an example text sequence [“that”, “is”, “a”, “question”]. Given “is”, assuming that the window size is 2, what the model needs to obtain is the probability of its neighboring words “that”, “a”, and “question”. At this time, “is” is called the head word, and other words are called background words. The basic idea of this model is to encode all words in One-Hot mode, input them into the neural network with only one hidden layer for training, and use the weight of hidden layer as the expression vector of words after training. The Skip-gram network structure is shown in Figure 1.

In this network, the dimensions of the input and output vectors of the neural network are the same, and there is no activation function in the hidden layer. In order to ensure that the output vector is a probability distribution, the output layer uses softmax. The number of neurons in the hidden layer depends on the dimension of word vector, and the number of neurons in the output layer is equal to the number of words in the corpus. Suppose the dictionary size is |v|. Match each word in the dictionary with integers from 0 to |v|−1 one by one, and establish the dictionary index set v = {0, 1, ..., |v|−1}. For any word in the dictionary, its corresponding integer in the dictionary is the index of the word. Assuming that a text sequence of length T is given, the word corresponding to time t is w_t, and the size of time window is m, then the Skip-gram model maximizes the probability of any central word generating background words, and the calculation method is shown in formula (1):

\[ J = \prod_{t=1}^{T} \prod_{m \leq j < m + 1} P(w_{t+j} \mid w_t). \]  

(1)

In order to maximize the objective function, the original maximum likelihood estimation becomes the minimization of formula (2).

\[ J = -\frac{1}{T} \sum_{t=1}^{T} \sum_{m \leq j < m + 1} \log \{ P(w_{t+j} \mid w_t) \}. \]  

(2)

v and u represent the vector of the central word and the background word, respectively. For the word with index i in the dictionary, the vector of the word as the central word and the background word is expressed as v_i and u_i, respectively. In order to embed the model parameters into the loss function, it is necessary to use the model parameters to
calculate the probability of generating background words from the center words in the loss function. Assume that the probabilities of generating background words from the central word are independent of each other. For the center word $w_c$ and the background word $w_o$, the probability that the center word in the loss function generates the background word is calculated by the softmax function.

$$
P_{wo|wc}(j) = \frac{\exp \left( u_i^T v_c \right)}{\sum_{i \in v} \exp \left( u_i^T v_c \right)}.
$$

When the sequence length is large, a smaller subsequence is usually randomly sampled to calculate the loss function, and the random gradient descent method is used to optimize the loss function. Then, the gradient of the generation probability is as follows:

$$
\frac{\partial \log P_{wo|wc}(j)}{\partial v_c} = u_o - \sum_{j \in v} \frac{\exp \left( u_i^T v_c \right)}{\sum_{i \in v} \exp \left( u_i^T v_c \right)} u_j.
$$

The formula is equivalent to the following:

$$
\frac{\partial \log P_{wo|wc}(j)}{\partial v_c} = u_o - \sum_{j \in v} P(w_j|w_c) u_j.
$$

After the gradient is calculated by this formula, the stochastic gradient descent method is used to update the model parameter $v_c$ iteratively. Similarly, the model parameter $v_o$ can be obtained. After the final training, for the word with index $i$ in the dictionary, two groups of word vectors $v_i$ and $u_i$ with this word as the central word and background word can be obtained.

2.2. Model Optimization. It can be found that the cost of gradient calculation in each step of the word vector model is related to the dictionary size $|v|$. When the dictionary is large in scale, ordinary training methods will consume a lot of space resources. Therefore, it is necessary to use approximate method to calculate the gradient, so as to reduce the calculation cost and improve the operation performance. The approximate training method adopted is sequence softmax. Sequence softmax uses a Huffman-coded binary tree to represent all words in the vocabulary. Each leaf node in the tree independently represents a word. There is a unique path from the root node to the leaf node for each leaf node. This path is used to estimate the probability of words represented by leaf nodes. Sequence softmax binary tree structure is shown in Figure 2.

Suppose $L(w)$ represents the number of nodes on the path from the root node to the leaf node. Let $n(w, i)$ be the $i$-th node on this path, and the vector of this node is expressed as $u_{n(w, i)}$. Then, the probability of word vector model generating word $w$ from arbitrary word $w_i$ is as follows:

$$
P(w|w_i) = \prod_{j=1}^{L(w)-1} \sigma \left( [n(w, j + 1) = \text{leftChild}(n(w, j))] \right) \cdot u_{n(w, j)}^T v_i,
$$

large in scale, ordinary training methods will consume a lot of space resources. Therefore, it is necessary to use approximate method to calculate the gradient, so as to reduce the calculation cost and improve the operation performance. The approximate training method adopted is sequence softmax. Sequence softmax uses a Huffman-coded binary tree to represent all words in the vocabulary. Each leaf node in the tree independently represents a word. There is a unique path from the root node to the leaf node for each leaf node. This path is used to estimate the probability of words represented by leaf nodes. Sequence softmax binary tree structure is shown in Figure 2.

Suppose $L(w)$ represents the number of nodes on the path from the root node to the leaf node. Let $n(w, i)$ be the $i$-th node on this path, and the vector of this node is expressed as $u_{n(w, i)}$. Then, the probability of word vector model generating word $w$ from arbitrary word $w_i$ is as follows:

$$
P(w|w_i) = \prod_{j=1}^{L(w)-1} \sigma \left( [n(w, j + 1) = \text{leftChild}(n(w, j))] \right) \cdot u_{n(w, j)}^T v_i,
$$

large in scale, ordinary training methods will consume a lot of space resources. Therefore, it is necessary to use approximate method to calculate the gradient, so as to reduce the calculation cost and improve the operation performance. The approximate training method adopted is sequence softmax. Sequence softmax uses a Huffman-coded binary tree to represent all words in the vocabulary. Each leaf node in the tree independently represents a word. There is a unique path from the root node to the leaf node for each leaf node. This path is used to estimate the probability of words represented by leaf nodes. Sequence softmax binary tree structure is shown in Figure 2.

Suppose $L(w)$ represents the number of nodes on the path from the root node to the leaf node. Let $n(w, i)$ be the $i$-th node on this path, and the vector of this node is expressed as $u_{n(w, i)}$. Then, the probability of word vector model generating word $w$ from arbitrary word $w_i$ is as follows:

$$
P(w|w_i) = \prod_{j=1}^{L(w)-1} \sigma \left( [n(w, j + 1) = \text{leftChild}(n(w, j))] \right) \cdot u_{n(w, j)}^T v_i,
$$

large in scale, ordinary training methods will consume a lot of space resources. Therefore, it is necessary to use approximate method to calculate the gradient, so as to reduce the calculation cost and improve the operation performance. The approximate training method adopted is sequence softmax. Sequence softmax uses a Huffman-coded binary tree to represent all words in the vocabulary. Each leaf node in the tree independently represents a word. There is a unique path from the root node to the leaf node for each leaf node. This path is used to estimate the probability of words represented by leaf nodes. Sequence softmax binary tree structure is shown in Figure 2.

Suppose $L(w)$ represents the number of nodes on the path from the root node to the leaf node. Let $n(w, i)$ be the $i$-th node on this path, and the vector of this node is expressed as $u_{n(w, i)}$. Then, the probability of word vector model generating word $w$ from arbitrary word $w_i$ is as follows:

$$
P(w|w_i) = \prod_{j=1}^{L(w)-1} \sigma \left( [n(w, j + 1) = \text{leftChild}(n(w, j))] \right) \cdot u_{n(w, j)}^T v_i,
$$

large in scale, ordinary training methods will consume a lot of space resources. Therefore, it is necessary to use approximate method to calculate the gradient, so as to reduce the calculation cost and improve the operation performance. The approximate training method adopted is sequence softmax. Sequence softmax uses a Huffman-coded binary tree to represent all words in the vocabulary. Each leaf node in the tree independently represents a word. There is a unique path from the root node to the leaf node for each leaf node. This path is used to estimate the probability of words represented by leaf nodes. Sequence softmax binary tree structure is shown in Figure 2.

Suppose $L(w)$ represents the number of nodes on the path from the root node to the leaf node. Let $n(w, i)$ be the $i$-th node on this path, and the vector of this node is expressed as $u_{n(w, i)}$. Then, the probability of word vector model generating word $w$ from arbitrary word $w_i$ is as follows:

$$
P(w|w_i) = \prod_{j=1}^{L(w)-1} \sigma \left( [n(w, j + 1) = \text{leftChild}(n(w, j))] \right) \cdot u_{n(w, j)}^T v_i,
$$

large in scale, ordinary training methods will consume a lot of space resources. Therefore, it is necessary to use approximate method to calculate the gradient, so as to reduce the calculation cost and improve the operation performance. The approximate training method adopted is sequence softmax. Sequence softmax uses a Huffman-coded binary tree to represent all words in the vocabulary. Each leaf node in the tree independently represents a word. There is a unique path from the root node to the leaf node for each leaf node. This path is used to estimate the probability of words represented by leaf nodes. Sequence softmax binary tree structure is shown in Figure 2.
where $\sigma$ represents sigmoid function and leftChild represents the left branch.

3. Machine Translation Model Integrating Attention Mechanism

The whole structure of the proposed A-BLSTM model is shown in Figure 3. Compared with other RNN models, one advantage of this model is that the number of model layers can change with the complexity of tasks. Moreover, with the increase of network layers, the number of nodes of A-BLSTM decreases layer by layer, and the computational complexity of each layer also decreases. We will take the three-tier network structure as an example to elaborate. Specifically, A-BLSTM includes the following three parts: sequence encoder based on BLSTM, local attention mechanism structure, and global attention mechanism structure.

3.1. Sequence Encoder Based on BLSTM. In RNN, the hidden layer state $h_t$ at time $t$ is calculated by a function $f$, as shown in formula (7):

$$h_t = \begin{cases} 
0, & t = 0, \\
 f(h_{t-1}, X_t), & \text{others},
\end{cases} \quad (7)$$

where $x_t$ represents the current input state and $f$ represents a nonlinear radiation transfer function.

LSTM is an improved time recursive neural network, which can effectively deal with the problem of long-term dependence in time series, so it has strong advantages in speech recognition. LSTM can effectively solve the problem that traditional RNN cannot learn long-distance dependence in the training process. LSTM contains a single memory cell unit, which will update its stored information as needed. Figure 4 shows the structure of LSTM network, in which the repeated modules represent the hidden layer in each iteration.

The LSTM unit at time $t$ is composed of a group of vectors, which include an input gate $i_t$, a forgetting gate $f_t$, an output gate $o_t$, a memory cell $c_t$, and a hidden state $h_t$. The conversion equation in LSTM network is as follows:
\[
\begin{align*}
i_i &= \sigma(W_iX_i + U_ih_{t-1} + b_i), \\
f_f &= \sigma(W_fX_i + U_fh_{t-1} + b_f), \\
o_o &= \sigma(W_oX_i + U_oh_{t-1} + b_o), \\
z_z &= \tanh(W_zX_i + U_zh_{t-1} + b_z), \\
c_c &= i_i \odot z_z + f_f \odot c_{c-1}, \\
h_h &= o_o \odot \tanh(c_c),
\end{align*}
\] (8)

where \( \odot \) represents the multiplication by element, \( b \) represents the bias vector parameter, and \( z_z \) represents an update gate.

Compared with unidirectional LSTM, bidirectional long short-term memory network (BLSTM) [27] uses extra backward information, which has the advantage of enhancing the memory ability of the network. The hidden state of each node in BLSTM can be calculated by the following formula.

\[
\tilde{h}_j^i = \left[ \begin{array}{c} \tilde{h}_j^i \\ \odot \tilde{h}_j^i \end{array} \right],
\] (9)

where \( \odot \) represents the connection operation, and \( \tilde{h}_j^i \) represents the output of BLSTM unit.

3.2. Local Attention Mechanism and Global Attention Mechanism. Traditional LSTM modeling uses the last hidden state as the sequence vector or averages to get the final sequence vector. However, the importance of different words or phrases in sentences is different [28]. Therefore, we introduce local attention mechanism into BLSTM. The mechanism of local attention is shown in Figure 5.

For the \( l \)-th layer in the network structure, \( r_l \) represents a representation vector of the input sequence, and its calculation formula is as follows:

\[
r_l = \sum_{j=1}^{D} Z_j \odot \tilde{h}_j^l,
\] (10)

\[
\begin{array}{|c|c|}
\hline
\text{Parameter type} & \text{Value} \\
\hline
\text{Glossary size} & 30000 \\
\text{Word vector dimension} & 512 \\
\text{Number of hidden layer nodes} & 512 \\
\text{Number of neural network layer} & 1, 2, 3, 4, \text{ and } 5 \\
\text{Learning rate} & 0.1 \\
\text{Dropout} & 0.5 \\
\text{Batch_size} & 128 \\
\text{Optimizer} & \text{Stochastic gradient descent} \\
\hline
\end{array}
\]
where $l \in [1, 3]$, $Z_l$ represents the normalized coefficient vector, and $D$ represents the length of the input sequence.

\[
Z_l = \sum_{j=1}^{D} \exp \left( u_l^j \right), \quad (11)
\]

\[
u_l^j = \tanh \left( W_u h_l^j + b_u \right),
\]

where $W_u$ and $b_u$ are parameters in the network.

The higher the number of layers of the network, the less original information retained in the obtained sentence vector, and the higher the abstract level of sentence meaning. Therefore, with the different input sequences and tasks, the trust weight of each layer in the network should be different. In order to reward the layer that is more meaningful for correctly classifying labels, we introduce the global attention mechanism in the network. The global attention mechanism is shown in Figure 6.

The principle of global attention mechanism is to give a weight to the classification probability of each layer, which represents how much probability the neural network trusts the output of this layer. The formula of global attention mechanism is as follows:

\[
p_l = P(\bullet|x_l; \theta) = \text{softmax} \left( W_p r_l + b_p \right),
\]

\[
o_l = \tanh \left( W_g p_l + b_g \right),
\]

\[
\alpha_l = \frac{\exp \left( o_l^g \right)}{\sum_{l} \exp \left( o_l^g \right)},
\]

\[
P = \sum_{l} \alpha_l p_l,
\]

where $W_p$, $W_g$, $b_p$, $b_g$, and $o_g$ are all parameters in the network. We can calculate an overall classification probability distribution $P$ through the classification probability of each layer and the trust weight of each layer in the network structure.

3.3. A-BLSTM. The proposed A-BLSTM model is a hierarchical structure, which continuously integrates the information of each layer in a bottom-up way and finally can effectively model the combined features. The layer number of A-BLSTM is expressed by $l \in [0, 3]$. For the $l$ layer, the input $h_l^j$ of the $j$-th node is calculated as follows:

\[
h_l^j = \left[ \tilde{h}_l^{j-1} \oplus \tilde{h}_l^{j-1} \right],
\]

\[
\tilde{h}_l^{j-1} = \text{BLSTM} \left( h_l^{j-1} \right),
\]

\[
\tilde{h}_l^{j+1} = \text{BLSTM} \left( h_l^{j+1} \right),
\]

where $\tilde{h}_l^{j-1}$ and $\tilde{h}_l^{j+1}$ are the outputs of BLSTM unit. A word is the input of the first layer in the network structure, and then, the output of each layer recursively becomes the input of the upper layer until the top layer of the network structure.

4. Experiment and Result Analysis

4.1. Machine Translation Evaluation Index. In order to evaluate the translation effect of the model, we use BLEU (bilingual evaluation under study) [29] proposed by IBM as the evaluation index of machine translation quality. BLEU is an internationally accepted evaluation method of machine translation. This method obtains the evaluation value by calculating the similarity between machine translation results and human translation results. The higher the similarity, the higher the score, that is, the higher the translation quality. The calculation formula of BLEU value is as follows:

\[
\text{BLEU} = \text{BP} \cdot \exp \left( \frac{1}{N} \sum_{n=1}^{N} \omega_n \log p_n \right),
\]

where $\text{BP}$ represents the penalty factor, $N$ represents the total number, $\omega_n$ represents the weight, and $p_n$ is the matching accuracy.

\[
\text{BP} = \begin{cases} 1, & l_c > l_r, \\ \exp \left( 1 - \frac{l_c}{l_r} \right), & l_c \leq l_r, \end{cases}
\]

where $l_c$ represents the length of the candidate translation and $l_r$ represents the effective length of the reference translation.
Figure 7: Continued.
4.2. Experimental Data and Data Processing. The experimental data is selected from the smaller data set of International Workshop on Spoken Language Translation (IWSLT). IWSLT is the most influential oral machine translation evaluation competition in the world. IWSLT 2015 data set includes 220,000 Chinese-English parallel sentences, including one development set data dev and three test set data (test1, test2, and test3). A variety of NMT models including A-BLSTM are built on TensorFlow, a deep learning framework. Parameters related to neural network are set as shown in Table 1.

During the experiment, firstly, preprocess the corpus: (1) segment the corpus—mainly for Chinese data, use Stanford word segmentation to divide Chinese sentences into words, as shown in Table 2; (2) symbol processing of corpus—mainly for English data, use tokenizer.per, a word segmentation script in Moses system, to insert spaces between words and punctuation marks in English data, as listed in Table 3; (3) converting uppercase into lowercase in English data; and (4) select the first 30000 high-frequency words from the processed training corpus and replace the rest words with < unk >.

4.3. Function Analysis of Network Layer Number. The influence of different layer number on the A-BLSTM model in Chinese-English machine translation tasks is shown in Figure 7. It can be seen that compared with the deeper or shallower hierarchical structure, the three-layer A-BLSTM structure has achieved the best results in both the development...
set and the test set. The performance of the two-layer A-BLSTM structure is poor, because only one abstract level of text vector representation can be obtained, so there is no advantage brought by applying the global attention mechanism. Generally speaking, with the increase of network layers, the network can learn more abstract semantic representation information, and the translation accuracy (BLEU value) will also be continuously improved. However, when there are too many network layers in the network structure (more than three layers), the translation accuracy will begin to decline. This is because for sequence modeling, too many layers are unnecessary, and extra noise will be introduced, which will reduce the accuracy of Chinese-English machine translation. In the follow-up experiments, A-BLSTM adopts three-layer architecture.

4.4. Analysis of the Effectiveness of Attention Mechanism. To illustrate the effectiveness of global attention mechanism, we compared the local attention BLSTM (LA-BLSTM) with A-BLSTM. The result of comparison between LA-BLSTM and A-BLSTM is shown in Figure 8. Obviously, the performance of A-BLSTM model is better than that of LA-BLSTM, especially on test2 data set and test3 data set. The experimental results show that the global attention mechanism can accurately identify which layer of the network structure’s representation vector is more reliable for translation.

4.5. Performance Comparison of Translation Models. Different neural networks are used for training and testing in the translation model, and the experimental results are shown in Table 4. In order to compare the translation performance of different models more intuitively, the data in Table 4 are displayed by histogram. The BLEU values of different translation models on the development set and the test set are shown in Figure 9.

From the results in Table 4 and Figure 9, we can find that the attention mechanism can improve the translation performance of NMT model and the proposed A-BLSTM model structure can effectively improve the machine translation performance by modeling both short sequences and long sequences. The main reason is that the global attention mechanism can determine which layer of representation vector in the input sequence should be given more attention (weight).

| Translation model | Dev | Test1 | Test2 | Test3 |
|-------------------|-----|-------|-------|-------|
| RNN               | 10.87 | 11.63 | 11.79 | 12.39 |
| LSTM              | 12.15 | 13.83 | 13.99 | 14.47 |
| GRU               | 12.16 | 13.85 | 14.01 | 14.49 |
| BiGRU             | 13.82 | 15.49 | 15.58 | 16.21 |
| Attention+RNN     | 14.75 | 16.41 | 16.62 | 20.34 |
| Attention+LSTM    | 20.74 | 22.47 | 22.53 | 23.27 |
| Attention+GRU     | 20.77 | 22.49 | 22.55 | 23.31 |
| Attention+BiGRU   | 21.86 | 23.37 | 23.41 | 24.28 |
| A-BLSTM           | 22.27 | 23.73 | 23.87 | 24.48 |
5. Conclusions

This paper proposes a neural machine translation model, A-BLSTM, which integrates attention mechanism. A hierarchical structure is adopted to represent the input sequence as a multilayer representation vector instead of a single fixed-length representation vector. The introduction of local attention mechanism can effectively select words or phrases with large amount of information in the input sequence. The introduction of global attention mechanism can effectively select a more credible intermediate representation vector. Experimental results show that the A-BLSTM model has achieved relatively high BLEU values in both Chinese and English machine translation tasks, which has surpassed other existing neural machine translation models. Subsequently, we will try to use quantum weak measurement to simulate the reader’s understanding of sentences from the perspective of quantum cognition, so as to further improve the translation performance.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. Brodbeck, A. Presacco, and J. Z. Simon, "Neural source dynamics of brain responses to continuous stimuli: speech processing from acoustics to comprehension," *NeuroImage*, vol. 172, no. 11, pp. 162–174, 2018.
[2] H. Choi, K. Cho, and Y. Bengio, "Fine-grained attention mechanism for neural machine translation," *Neurocomputing*, vol. 284, no. 4, pp. 171–176, 2018.
[3] B. Erik, H. Xiang, and L. Meng, "Does machine translation affect international trade? Evidence from a large digital platform," *Social Science Electronic Publishing*, vol. 12, no. 8, pp. 29–42, 2018.
[4] M. S. Kumar, D. Dipankar, and B. Sivaji, "MTII2017: machine translation using recurrent neural network on statistical machine translation," *Journal of Intelligent Systems*, vol. 36, no. 1, pp. 3–19, 2018.
[5] J. Su, J. Zeng, D. Xiong, Y. Liu, M. Wang, and J. Xie, "A hierarchy-to-sequence attentional neural machine translation model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 3, pp. 623–632, 2018.
[6] Z. Tan, J. Su, B. Wang, Y. Chen, and X. Shi, "Lattice-to-sequence attentional neural machine translation models," *Neurocomputing*, vol. 284, no. 4, pp. 138–147, 2018.
[7] J. Huang, Y. Sun, W. Zhang, H. Wang, and T. Liu, "Entity highlight generation as statistical and neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1860–1872, 2018.
[8] V. Mann and V. Venkatasubramanian, "Retrosynthesis prediction using grammar-based neural machine translation: an information-theoretic approach," *Computers & Chemical Engineering*, vol. 23, no. 1, pp. 23–41, 2021.
[9] A. Rishab, K. Anupam, M. Deepak, and P. Sanidhya, "Employing differentiable neural computers for image captioning and neural machine translation," *Procedia Computer Science*, vol. 173, pp. 234–244, 2020.

[10] S. Bayati, S. Kurnaz, and A. Ali, "Unsupervised weighting of transfer rules in rule-based machine translation using maximum-entropy approach," *Journal of Information Science and Engineering*, vol. 36, no. 2, pp. 309–322, 2020.

[11] A. Matheus, C. P. Adriano, and B. Fabricio, "A comparative study of machine translation for multilingual sentence-level sentiment analysis," *Information Sciences*, vol. 512, pp. 1078–1102, 2020.

[12] S. Berrichi and A. Mazroui, "Addressing limited vocabulary and long sentences constraints in English–Arabic neural machine translation," *Arabian Journal for Science and Engineering*, vol. 15, no. 1, pp. 58–68, 2021.

[13] M. Dasa, H. Petr, M. Michal, and S. Ján, "Evaluation of machine translation quality through the metrics of error rate and accuracy," *Procedia Computer Science*, vol. 171, pp. 1327–1336, 2020.

[14] A. Solyman, Z. Wang, and Q. Tao, "Synthetic data with neural machine translation for automatic correction in Arabic grammar," *Egyptian Informatics Journal*, vol. 66, no. 12, pp. 216–223, 2020.

[15] J. Zhang, M. Utiyama, E. Sumita, and N. Graham, "Improving neural machine translation through phrase-based soft forced decoding," *Machine Translation*, vol. 34, no. 1, pp. 21–39, 2020.

[16] M. K. Vathsala and G. Holi, "RNN based machine translation and transliteration for twitter data," *International Journal of Speech Technology*, vol. 23, no. 10, pp. 538–551, 2020.

[17] H. Shan, Y. Zhang, Q. Yang et al., "3-D convolutional encoder-decoder network for low-dose CT via transfer learning from a 2-D trained network," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1522–1534, 2018.

[18] X. W. Jiang, B. Hu, Z. H. Guan, X. H. Zhang, and L. Yu, "Best achievable tracking performance for networked control systems with encoder-decoder," *Information Sciences*, vol. 305, pp. 184–195, 2015.

[19] L. Su and L. Zhou, "Exponential synchronization of memristor-based recurrent neural networks with multiproportional delays," *Neural Computing and Applications*, vol. 31, no. 5, pp. 27–35, 2019.

[20] P. A. Peng, A. Wz, and Z. A. Yi, "Cost sensitive active learning using bidirectional gated recurrent neural networks for imbalanced fault diagnosis," *Neurocomputing*, vol. 407, pp. 232–245, 2020.

[21] B. A. Sangeroki and T. W. Cenggoro, "A fast and accurate model of thoracic disease detection by integrating attention mechanism to a lightweight convolutional neural network," *Procedia Computer Science*, vol. 179, no. 11, pp. 112–118, 2021.

[22] C. Gulcehre, O. Firat, K. Xu, K. Cho, and Y. Bengio, "On integrating a language model into neural machine translation," *Computer Speech & Language*, vol. 45, no. 9, pp. 137–148, 2017.

[23] A. Sano, W. Chen, D. Lopez-Martinez, S. Taylor, and R. W. Picard, "Multimodal ambulatory sleep detection using LSTM recurrent neural networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1607–1617, 2019.

[24] S. Wu, D. Zhang, Z. Zhang, N. Yang, M. Li, and M. Zhou, "Dependency-to-dependency neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2132–2141, 2018.

[25] Q. Ren, L. S. Yi, and W. L. Wan, "Research on the LSTM Mongolian and Chinese machine translation based on morpheme encoding," *Neural Computing and Applications*, vol. 32, no. 1, pp. 41–49, 2020.

[26] G. R. Devi, P. V. Veena, M. A. Kumar, and K. P. Soman, "Entity extraction for Malayalam social media text using structured skip-gram based embedding features from unlabeled data," *Procedia Computer Science*, vol. 93, pp. 547–553, 2016.

[27] A. Mahmoud and M. Zrigui, "BLSTM-API: bi-LSTM recurrent neural network-based approach for Arabic paraphrase identification," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 4163–4174, 2021.

[28] L. Yu, X. Yu, and Y. Zhang, "Microinstrument contact force sensing based on cable tension using BLSTM–MLP network," *Intelligent Service Robotics*, vol. 13, no. 1, pp. 123–135, 2020.

[29] F. A. Burns, P. Villiers, B. Z. Pearson, and T. B. Champion, "Dialect-neutral indices of narrative cohesion and evaluation," *Language Speech and Hearing Services in Schools*, vol. 43, no. 2, pp. 132–152, 2012.