Hindawi
Mobile Information Systems
Volume 2021, Article ID 9985251, 10 pages
https://doi.org/10.1155/2021/9985251

Research Article

Efficient English Translation Method and Analysis Based on the Hybrid Neural Network

Chuncheng Wang

Tongling University, Tongling 244061, China

Correspondence should be addressed to Chuncheng Wang; chad825@tlu.edu.cn

Received 29 March 2021; Accepted 4 May 2021; Published 15 May 2021

Academic Editor: Jianhui Lv

Neural machine translation has been widely concerned in recent years. The traditional sequential neural network framework of English translation has obvious disadvantages because of its poor ability to capture long-distance information, and the current improved framework, such as the recurrent neural network, still cannot solve this problem very well. In this paper, we propose a hybrid neural network that combines the convolutional neural network (CNN) and long short-term memory (LSTM) and introduce the attention mechanism based on the encoder-decoder structure to improve the translation accuracy, especially for long sentences. In the experiment, this model is implemented based on TensorFlow, and the results show that the BLEU value of the proposed method is obviously improved compared with the traditional machine learning model, which proves the effectiveness of our method in English-Chinese translation.

1. Introduction

The machine translation is an approach that uses computers to automatically convert different languages, which is an important research field of natural language processing (NLP) and artificial intelligence (AI) [1]. It is also one of the most common services on the internet. Although it is still challenging to make the translation quality of machine translation reach the level of professional translators, machine translation has obvious advantages in translation speed in some cases, such as the translation quality is not concerned significantly, or in specific domain translation tasks [2, 3]. In the view of the complexity and application of machine translation, this field is regarded as a key research direction, and it has become one of the most active research fields in natural language processing.

It should be mentioned that neural machine translation (NMT) is the most popular machine translation method that loads specific algorithms into the neural model framework and uses the node-to-node method to optimize the model [4]. At present, with the conditions of large-scale corpus and computing power, NMT has shown great potential and has developed into a new machine translation method. This method only requires bilingual parallel corpus and is convenient for training large-scale translation models. It not only has high research value but also has strong industrialization capabilities. In many language pairs, NMT has gradually surpassed phrase statistical machine translation. Junczys-Dowmunt et al. [5] used the United Nations Parallel Corpus v 1.0 to compare NMT and phrase statistical machine translation on 30 language pairs. It can be found that NMT surpassed phrase statistical machine translation on 27 language pairs. In addition, on tasks related to Chinese, such as Chinese-English translation tasks, NMT has 6–9 higher BLEU (Bilingual Evaluation Understudy) values. At the Workshop on Machine Translation (WMT) in 2016 [6], the NMT system developed by the University of Edinburgh surpassed phrase-based and syntactic-based statistical machine translation in English-to-German translation tasks. Furthermore, in the industry, Google Translate has adopted NMT instead of statistical machine translation in some languages to provide external services [7]. The well-known

commercial machine translation company SYSTRAN has also developed a corresponding NMT system, covering 12 languages and 32 language pairs.

NMT has an advantage that it can use neural networks to achieve direct translation from the source language to the target language [8]. This translation idea can be traced back to the 1990s; some scholars used small-scale corpus to implement a neural network-based translation method. Due to the limitation of corpus resources and computing power, it did not receive corresponding attention. After the rise of the deep learning boom, neural networks are often used for statistical machine translation language models, word alignment, translation rule extraction, and so on. Until 2013, Blunsom and Kalchbrenner reproposed the neural network-based translation method which shows great application potential [9]. Subsequently, Sutskever, Cho, Jean, and others, respectively, realized the corresponding machine translation model based on the neural network [10]. These are classic NMT models, which are essentially sequence-to-sequence models. They can be used not only for machine translation but also for question-answering systems, text summarization, and other natural language processing tasks. Different from the discrete representation method of statistical machine translation, NMT uses a continuous space representation method to represent words, phrases, and sentences. In translation modeling, the necessary steps of statistical machine translation such as word alignment and translation rule extraction are not required, and the neural network is used to complete the mapping from the source language to the target language [11]. Additionally, the other is called the encoder-decoder model, in which the encoder reads the source language sentence and encodes it into a vector with fixed dimensions, and the decoder reads the vector and sequentially generates the target language word sequence. The encoder-decoder model is a general framework that can be implemented by different neural networks, such as long short-term memory (LSTM) neural networks [12] and gated recurrent neural networks (GRNNs) [13]. NMT has been verified that its translation effect is close to or equal to the phrase-based statistical machine translation method. It also has great advantages in some fine-grained evaluation indicators of translations. For example, focusing on the English-to-German translation evaluation task in the 2015 International Workshop on Spoken Language Translation (IWSLT), Bentivogli et al. made a detailed comparative analysis of the translations of phrase statistical machine translation and NMT [14]. As a result, in the neural machine translation, morphological errors were reduced by 19%, vocabulary errors were reduced by 17%, and word ordering errors were reduced by 50%. Among the word order errors, the verb order errors were reduced by 70%.

The current mainstream of NMT is combining with the encoder-decoder structure, and the algorithm is connected through the attention mechanism between the encoder and the decoder [1]. However, NMT based on the structure of the encoder and decoder is a general model, which is not specifically designed for the machine translation task. This leads to some problems including two aspects: firstly, although neural machine translation is a major improvement of the attention mechanism, its disadvantage is that historical attention information is not considered when generating the target language words, and the constraint mechanism is weak. Furthermore, in some cases, it is not necessary to pay too much attention to the source language information when generating the target language words. For example, when generating the function word "The" in Chinese-English translation, much attention should be paid to the relevant information of the target language [15]. In addition to the above, overtranslation and undertranslation problems may occur in NMT, and it is also necessary to improve the existing attention mechanism [16, 17]. In summary, the attention mechanism optimization is a hot and difficult point in NMT research.

It should be noted that the attention mechanism is a classic neural machine translation model, which improves the representation of the source language and generates source language-related information dynamically in the decoding process to improve the translation effect [18]. Attention-based NMT encodes the source language sentence into a vector sequence instead of a fixed vector. When generating the target language word, it can use the source language word information related to the generation of the word, and the corresponding word can be in the source language. The bilingual vocabulary correspondence realized by the attention mechanism is called soft alignment. Compared with the hard alignment method of statistical machine translation, this method does not limit the alignment length of the target language words and the source language words and can avoid the air-to-air problem in the hard alignment method [19]. However, the attention mechanism has the problem of a large amount of calculation. In order to reduce the amount of calculation, Xu et al. [20] divided attention into soft attention and hard attention on the task of image description generation. The former refers to assigning weights to all regions of the original image, and the amount of calculation is relatively large. And the latter refers to only paying attention to part of the original image area, which can reduce the computational complexity. Based on the above ideas, Luong et al. [21] proposed a local attention model, which is an improvement on global attention. When calculating the context vector c_t, global attention needs to consider all the coding sequences of the source language. Local attention only needs to focus on a small context window in the source language coding, which can significantly reduce the complexity of computation. The core of this method is to find an alignment position related to the generated word from the source language. Local attention only focuses on a small part of the source language when generating the context vector and filters out irrelevant information, which is suitable for long sentence translation. In the WMT 2014 English to German translation, local attention increased by 0.9 BLEU value compared to global attention. In the long sentence translation experiment, as the sentence length increases, the local attention method does not reduce the translation quality. In addition, on the English-German word alignment corpus of the Aachen University of Technology, the local attention word alignment error rate was 34%, and the global attention word alignment

error rate was 39%. In particular, the supervised attention mechanism is the hot topic in this field that uses high-quality prior word alignment knowledge to guide the attention mechanism. Liu et al. proposed a method of using statistical machine translation word alignment information as a priori knowledge to guide the attention mechanism [22]. This method uses GIZA++ to obtain the word alignment information of the training corpus, and then, in the model training, statistical machine translation word alignment is used as a priori knowledge to guide the attention mechanism so that the attention-based word alignment is possible for statistical machine translation. Finally, no prior word alignment information is needed during the test. The experiment uses the Chinese-English machine translation evaluation corpus organized by the National Institute of Standards and Technology (NIST) in 2008. Compared with the attention-based neural machine translation, this method improves the BLEU value by 2.2. On the Tsinghua word alignment corpus, the word alignment error rate of GIZA++ is 30.6%, the word alignment error rate based on attention neural machine translation is 50.6%, and the word alignment error rate of this method is 43.3%. It can be seen that there is a supervision mechanism that can significantly improve the word alignment quality of the attention mechanism, but there is still a big gap compared with the statistical machine translation word alignment, and the attention mechanism still needs improvement.

Aiming at the above disadvantages of the traditional encoder-decoder model framework, this paper proposes an English-Chinese translation model based on a hybrid neural network and an improved attention mechanism. The main idea of the method is to combine the attention mechanism with the neural network to train the local attention of the translation model. Compared with traditional machine learning methods such as least square support vector machine (LSSVM) and extreme learning machine, deep learning methods, i.e., LSTM and convolutional neural network (CNN), have more powerful learning capabilities and good approximation capabilities for the text data in processing regression problems. Therefore, this paper mixes these two networks to improve the ability of the translation model to connect to the context, thereby improving the translation quality of the model.

The rest of this paper is organized as follows. Section 2 presents the detail of the encoder-decoder structure model, CNN, RNN, and attention mechanism. Section 3 presents the hybrid neural network with an improved attention mechanism proposed in our work. Experimental results and discussion are reported in Section 4. Finally, the conclusion of this paper is given in Section 5.

2. Materials and Methods

2.1. Encoder-Decoder Structure Model. The encoder-decoder structure designed in this paper is the core part of the machine translation model, which is composed of an encoder and decoder. The encoder transforms the input data of the neural network into a fixed length of data. The decoder reversely decodes the data and then outputs the translated sentences, which is also the basic idea of the sequence model. The main process is shown in Figure 1.

The model of encoder-decoder consists of three parts: the input x, the hidden state h, and the output y. The encoder reads the input $x = (x_1, x_2, \ldots, x_i)$ and transforms the code into the hidden state $h = (h_1, h_2, \ldots, h_i)$ when adopting the RNN:

$$\begin{aligned} h_i &= f(x_i, h_{i-1}), \\ c &= q(\{h_1, \ldots, h_i\}), \end{aligned} \tag{1}$$

where c is the sentence representation in the source language and f and q are nonlinear functions. The decoder can generate target language words with the given source language representation c and the precursor output sequence $\{y_1, \ldots, y_{t-1}\}$; the definition is as follows:

$$p(y) = \prod_{t=1}^{T} p(y_t | \{y_1, \ldots, y_{t-1}\}, c), \tag{2}$$

where $y = (y_1, y_2, \ldots, y_T)$, and when using the RNN,

$$p(y_t | \{y_1, \ldots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c). \tag{3}$$

In this model, g is a nonlinear function that is used to calculate the probability of y_t, and s_t is the hidden state of the RNN, $s_t = f(s_{t-1}, y_{t-1}, c)$. The encoder and decoder can be trained jointly in the following form:

$$L(\theta) = \max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log\, p_\theta(y_n | x_n). \tag{4}$$

(x_n, y_n) is the bilingual sentence pair, and θ is a parameter of the model, which can be calculated by the gradient descent method.

2.2. Convolutional Neural Network (CNN). CNN is a special deep learning neural network, which is often used to process data with known grid topology [20]. It is widely used in time series analysis, computer vision, and NLP. According to different data streams, the CNN can be divided into one-dimensional convolution, two-dimensional convolution, and three-dimensional convolution. One-dimensional convolution is widely used in time series analysis and natural language processing. The CNN structure adopted in this paper belongs to the one-dimensional convolution neural network [20], as shown in Figure 2.

Each kind of CNN consists of an input layer and output layer and the core operation part, i.e., convolution layer, pooling layer, and full connection layer. In one-dimensional convolution, the function of convolution can be understood as extracting the translation features of the data in a certain direction. In our work, the essence of the convolution

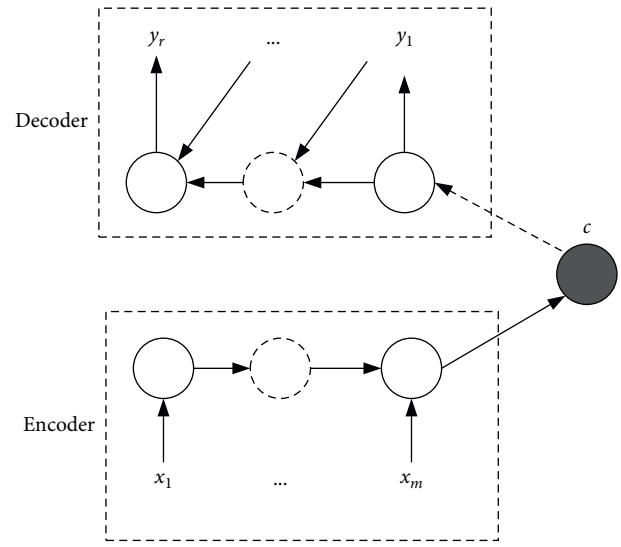

Figure 1: Encoder-decoder translation flowchart.

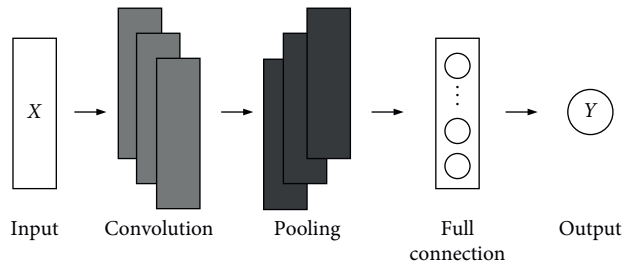

Figure 2: One-dimensional convolutional neural network.

operation is cyclic product and addition, and its mathematical expression is as follows:

$$y(k) = h(k) * u(k) = \sum_{i=0}^{N} h(k-i)u(i), \tag{5}$$

where y, h, and u are time series, k denotes the convolution number, and N is the length of u.

The basic architecture of the general CNN encoder is shown in Figure 3, and its fixed architecture consists of the following six layers:

Layer 0: input layer that uses the embedded vector form of words and sets the maximum length of the sentence to 40 words. For the sentences shorter than this vector, zero padding is often placed at the beginning of the sentence.

Layer 1: the convolution layer after layer 0, whose window size is 3. The boot signal is injected into the layer as the "boot version."

Layer 2: the local gating layer after the first layer, and it only makes the weighted sum of the feature graph for the nonadjacent window of size 2.

Layer 3: the convolution layer after layer 2, which performs another convolution, and its window size is 3.

Layer 4: this layer performs global gating on the function diagram on layer 3.

Layer 5: fully connected weight, which maps the output of layer 4 to this layer as the final representation.

As is shown in Figure 3, convolution in layer 1 moves on the sliding window of the word, and a similar definition of the window continues to a higher level. Formally, for the source sentence input $x = \{x_1, \ldots, x_N\}$, the convolution unit of F-type feature mapping on layer L is shown in the following equation:

$$z_i^{(l,f)}(x) = \sigma\left(\omega^{(l,f)}\hat{z}_i^{l-1} + b^{(l,f)}\right), \quad l = 1, 3, \ f = 1, 2, \ldots, F_l, \tag{6}$$

where $z_i^{(l,f)}(x)$ gives the output of position i in layer L whose feature map type is f; $\omega^{(l,f)}$ is the parameter of f in layer L; $\sigma(\cdot)$ is the activation function of sigmoid; and $\hat{z}_i^{l-1}$ denotes the convolution segmentation at position i of layer 1, and $\hat{z}_i^0 = \{x_i^T, x_{i+1}^T, x_{i+2}^T\}$ to connect the vector of 3 words from the sentence input.

2.3. Recurrent Neural Network (RNN). The encoder-decoder framework is a part of the neural network, and it is necessary to build an appropriate neural network model to run the framework. RNN is the most widely used among many neural network models, which is a variant model of the feedforward neural network. Its main feature is to process different length data series. Figure 4 shows the structure of the RNN, in which the network has recursive property, and the state of each time has a great relationship with the previous activation state. In this figure, $x = \{x_1, x_2, \ldots, x_T\}$ denotes the variable length sequence data, and at each time point t, the hidden state h_t is updated by the following formula:

$$h_t = f(h_{t-1}, x_t), \tag{7}$$

where f is a nonlinear function. We map the input x, and y is the target sequence of the model which is usually given by the training corpus. L is the loss function; U is the weight matrix input to the hidden layer; W is the weight matrix from the hidden layer to the hidden layer; V is the weight matrix from the hidden layer to the output, and t is the weight matrix from the hidden layer to the output, which has the range [1, T]. The whole network is updated as follows:

$$\begin{aligned} a_t &= \mathrm{W}h_{t-1} + \mathrm{U}x_t + b, \\ h_t &= \tanh(a_t), \\ o_t &= \mathrm{V}h_t + c, \\ \hat{y}_t &= \mathrm{soft\,max}(o_t). \end{aligned} \tag{8}$$

The RNN makes different length sequences having the input vectors with the same dimension and the same transformation function, and parameters can be used at each time point, which are more suitable for processing variable length sequence data. In addition, the loop structure can capture all the precursor states in theory, which solves the problem of long-distance dependence to a certain extent.

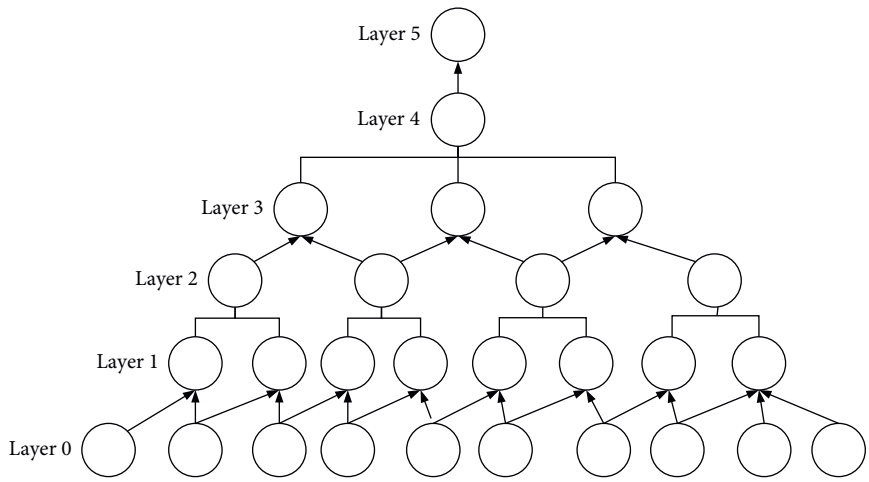

Figure 3: CNN coding illustration.

RNN processes variable length sequences by circulating hidden state units. However, gradient vanishing and gradient explosion may occur in RNN training because it is difficult for the RNN to capture the long-term dependence of the data. Thus, long-term and short-term memory (LSTM) network is proposed to solve the problem of RNN gradient disappearing [23]. LSTM is a kind of gated recurrent neural network, which is a special form of the RNN and can capture the long-term dependence between data.

There are three gates in the LSTM cell structure: forgetting gate, input gate, and output gate, and the structure is shown in Figure 5. In LSTM, long-term memory or forgetting information is realized through the input gate, forgetting gate, output gate, and memory unit. If the current time is t, the current input state information and the output value of the current LSTM are the memory unit state of the previous time. The calculation formula of the LSTM cell is shown in equations (9)–(14):

$$\Gamma_f = \sigma\left(W_f\left[h^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_f\right), \tag{9}$$

$$\Gamma_i = \sigma\left(W_i\left[h^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_i\right), \tag{10}$$

$$\Gamma_o = \sigma\left(W_o\left[h^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_o\right), \tag{11}$$

$$\widehat{s}^{\langle t\rangle} = \tanh\left(W_s\left[h^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_s\right), \tag{12}$$

$$s^{\langle t\rangle} = \Gamma_f s^{\langle t\rangle} + \Gamma_i \widehat{s}^{\langle t\rangle}, \tag{13}$$

$$h^{\langle t\rangle} = \Gamma_0 \tanh\left(s^{\langle t\rangle}\right), \tag{14}$$

where W is the weight vector for each gate; b is the bias vector; σ is the sigmoid function; and tanh is a nonlinear activation function.

Furthermore, gated recurrent unit (GRU) is the variant of LSTM, which has simpler memory units. This structure combines the input gate and forgetting gate of the long- and short-term memory cycle into the update gate and then introduces the reset gate. It uses the update gate to control the history information and new information to be forgotten in the current state and uses the reset gate to control the information quantity obtained from historical information in the candidate state. As shown in Figure 6, the GRU memory unit has only two gates: reset door and update door. The reset gate r_t controls the degree of status information of the previous moment, and the update gate z_t determines the quantity of the memory reserved in front. The simple memory unit of the GRU makes its parameters less than LSTM, and its performance is equivalent to or even better than LSTM. The calculation formula of the reset gate and update gate is as follows:

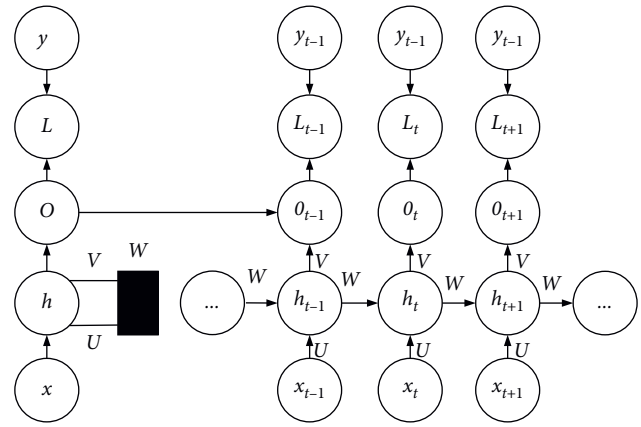

Figure 4: Structure diagram of the RNN.

$$\begin{cases} z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right] + b_z\right), \\ r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right] + b_r\right), \\ \overline{h}_t = \tanh\left(W_a\left[r_t * h_{t-1}, x_t\right] + b_a\right), \\ h_t = \left(1 - z_t\right) * h_{t-1} + z_t * \overline{h}_t, \end{cases} \tag{15}$$

where W_z, W_r, and W_a are weight matrices and b_z, b_r, and b_a are deviation vectors.

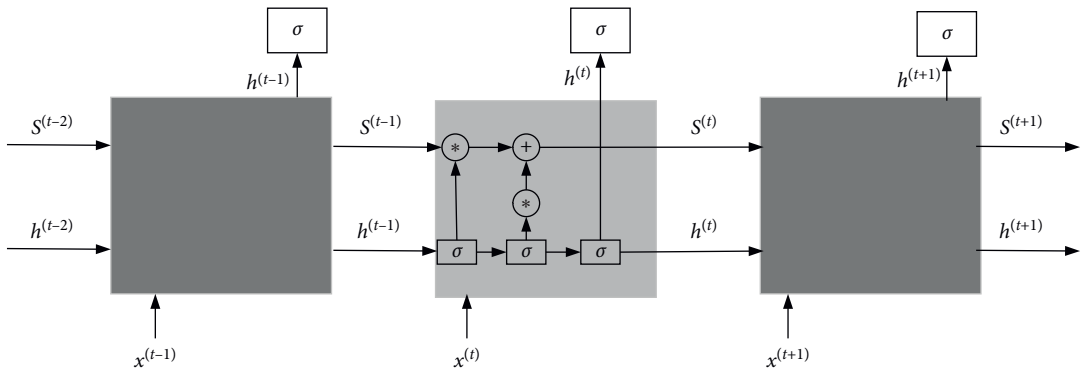

Figure 5: LSTM structure.

2.4. RNN with the Attention Mechanism. When the human brain observes an object, it often focuses on some parts, and these parts are also the key to obtain information from things. This information has a strong guiding role in cognition of similar things, and the attention mechanism is designed to imitate this cognitive process. The application of the attention mechanism in computer vision and natural language processing has achieved good results. This paper applies the attention mechanism to the analysis of text series.

In the analysis of the text sequence, the CNN is used to extract the spatial features of the sequence. As too many or nonkey features will affect the final prediction results after LSTM is used to extract the spatial and temporal features, the attention mechanism is used to extract the key features of the sequence. The attention mechanism is similar to a weighted summator or a key feature extractor, which mainly performs the weighted summation operation. The attention model proposed in this paper is shown in Figure 7, and the vector c is the key feature to be extracted, as is shown in the following equation:

$$c = \sum_{i=1}^{m} \beta_i v_i, \tag{16}$$

where m is the time step sum of the input of the LSTM network; v is the output eigenvector of the LSTM network; and β is the weight of vector v. To obtain the weight β, we add a small neural network a to the attention model, and the output layer activation function is softmax. The calculation is shown as follows:

$$\beta_i = \frac{\exp(e_i)}{\sum_{k=1}^{m} \exp(e_k)}, \tag{17}$$

where e_i can be computed in $e_i = a(v_i) = \sigma(W v_i + b)$ and σ is the sigmoid function. W is the weight matrix from the input layer to the hidden layer in Figure 7, and b is the offset value matrix.

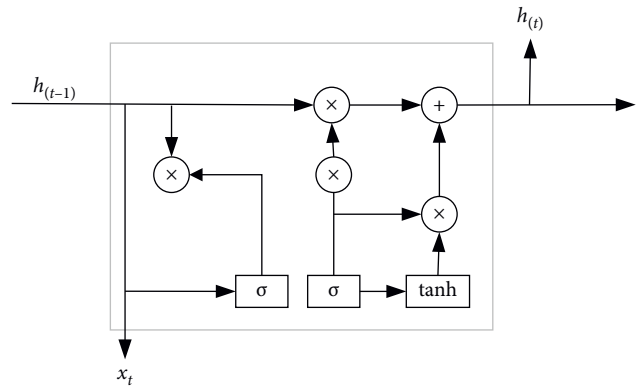

Figure 6: GRU structure.

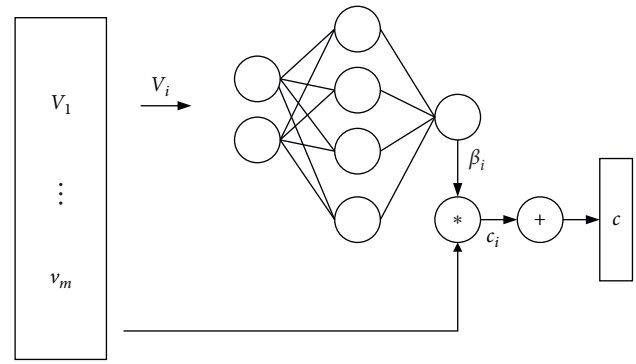

Figure 7: Attention mechanism model.

3. Proposed Method

In order to reduce the gradient problem caused by the long data sequence, a popular method is to combine the RNN with encoder-decoder. However, the operational performance of the encoder-decoder framework is limited to a certain extent, which results in that the long input data have high computational complexity. The attention mechanism can solve the problem of limited operation by establishing the decoder and sentence segment transmission channel, i.e., the decoder can return to view the input data at any time. In this way, the intermediate data can be omitted; thus, the operation performance will be improved, and the translation accuracy will be improved.

The left-right sequential operation mechanism of the RNN may lead to the limitation of the parallel operation ability of the model and data module loss. The attention mechanism is helpful to solve the above problems because the data distance of any position in the translation data can

be changed to 1, which improves the parallelism of the model but depends on the influence of the previous sequence operation no longer.

The main process of the attention mechanism consists of four steps: (1) weighting the input data of the neural network and importing them into the encoder; (2) importing the data into the decoder; (3) the decoder queries the data weight in the decoding process as the reverse input data; (4) computing the weighted average value of the data in each state. The simplified implementation process of the attention mechanism is shown in Figure 8.

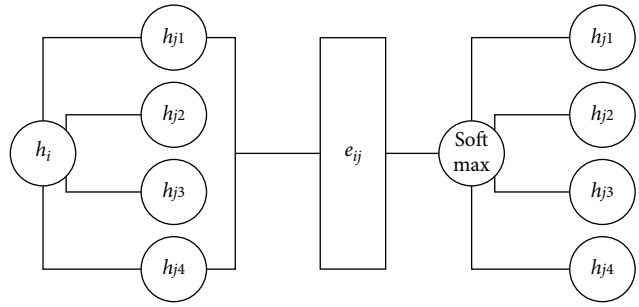

Figure 8: Schematic diagram of the attention mechanism.

3.1. Model Framework. In order to highlight the advantages of the attention mechanism, this paper introduces the attention mechanism into the model framework of the RNN and realizes the translation task by establishing the encoder-decoder framework. The neural connection part of the model is realized by the attention mechanism, which is helpful to take advantages of the attention mechanism.

Figure 9 shows the attention mechanism model constructed in this paper, whose overall structure is composed of the encoder and decoder. The encoder is composed of a single-layer structure and a single-layer structure of the precoding network, and the block number is N_c. The structure of the decoder is similar to that of the encoder, which is also composed of N_c blocks, but has no head attention layer. The neural network in this paper uses differential network connection, and the remarkable feature of this approach is that the network has entered the standard level for data processing.

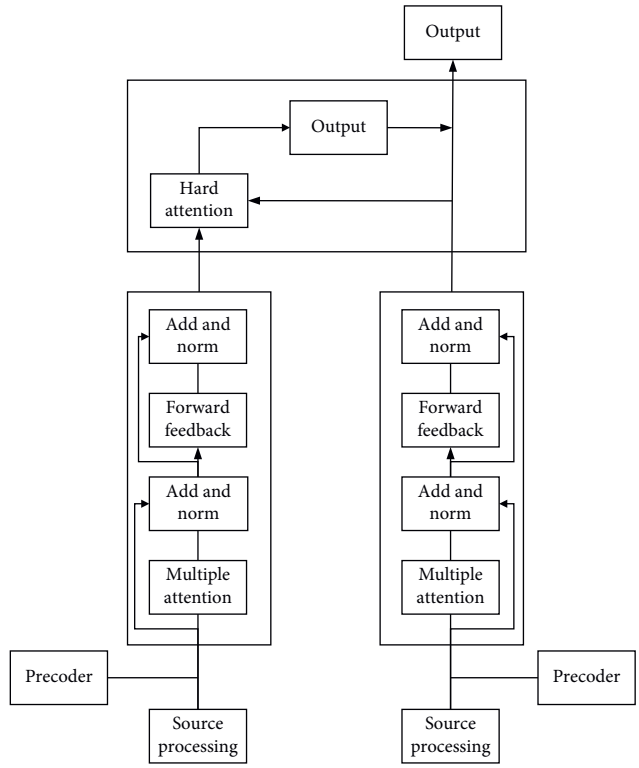

Figure 9: Model frame structure.

3.2. Attention Mechanism Module. The attention mechanism module is mainly divided into the encoder module and decoder module. The input part of the encoder module is the whole data sequence, and three input matrices are used in this part, i.e., Q, K, and V. The attention mechanism function can be regarded as a mapping relationship as follows:

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{18}$$

The calculation process of the attention mechanism is as follows:

(1) The sentence data to be translated are weighted by three different matrices, i.e., Q, K, and V, and each sentence will get the three vectors

(2) The weight distribution of the above three matrices is calculated by scaling the dot product to get a numerical value

(3) Take the weight value in Step (2) as the activation function

(4) Multiply the output of Step (3) with the V matrix to get the final result

4. Experiment and Analysis

All the experiments are carried out on the Alibaba cloud server ECS, whose CPU type is Intel Skylake Xeon Platinum 8163 with 2.5 GHz, and the memory is 8 GB. All the program codes are written in Python (version 3.7.7) in TensorFlow [24].

4.1. Data Acquisition and Evaluation Methods.

(1) Data acquisition: the training data of this paper are extracted from LDC data [25]. Only the part of the source pair which is less than 40 words in length is reserved, which covers more than 90% of the sentences. The bilingual training data consist of 221k

sentence pairs, including 5 million Chinese words and 6.8 million English words. After length-limited filtering, the development set is NIST MT03, which contains 795 sentences, and the test set is MT04 and MT05, which contain 1499 sentences and 917 sentences, respectively.

(2) Preprocessing: use Giza++ to achieve word alignment in two directions with the "growth-determine-final" and "balance strategy" on the corpus [23]. The improved Kneser–Ney smoothing 4-gram language model is trained on the new Chinese part of English Gigaword corpus, which contains 306 million words, by using the SRI language modeling toolkit. Then, the Chinese sentence is parsed into a mapping dependency tree using Stanford Parser.

(3) Neural network optimization: when training the neural network, the source words and target words are limited to the most common 20k words in Chinese and English, covering 97% and 99% of the words in the two corpora, respectively. All vocabularies and words are mapped to the special token UNK. Random gradient descent is used to train the joint model, and the minimum batch size is set to 500. All joint models use 3-word target (4-gram LM). The final representation of the CNN encoder is a vector with a size of 100. The final DNN layer of the joint model is a standard multilayer perceptron, and the top layer has softmax.

(4) Evaluation: in order to evaluate the prediction error, BLEU value is used as the evaluation index as follows:

$$\begin{aligned} \text{BLEU} &= \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log P_n\right), \\ \text{BP} &= \begin{cases} 1 & \text{if } c > r, \\ e^{1-r/c} & \text{if } c \le r. \end{cases} \end{aligned} \tag{19}$$

It should be noted that the BLEU value is calculated for a translation sample.

4.2. Experimental Test and Result Analysis. The steps of the experiment are as follows:

(1) The corpus is processed to cut long sentences into words

(2) The words are numbered and stored as files, and the files are stored in a PC

(3) Normalizing the text, making up for the insufficient sentence length, and intercepting the excessive sentence length

(4) Training the processed sentences, and then the BLEU value is evaluated

In the experiment, the comparison test is used to evaluate the error of single LSTM [26], LSSVM [27], CNN [28], CNN-LSTM [29] without the attention mechanism, LSTM with the attention mechanism [30], and the proposed model. The comparison results are shown in Table 1.

It can be seen from Table 1 that LSTM's translation effect has been significantly improved by 2.8 and 7.36 BLEU after combining the CNN and attention mechanism with the same settings. Furthermore, LSTM is higher than CNN's average values of 2.5 and 0.49 BLEU in MT04 and MT05, respectively. The results indicate that LSTM and attention mechanism can provide discriminative information in decoding. It is worth noting that CNN + LSTM is more informative than LSSVM. Therefore, it is speculated that this is due to the following two facts:

(1) CNN + LSTM avoids the spread of errors and pseudo-effects in the learned word alignment

(2) Guidance signals in CNN + LSTM provide supplementary information to evaluate translation

In addition, the reason CNN can get high gain on BLEU is that it encodes the whole sentence, and the representation should be far away from the best representation of the joint language model. Therefore, as the CNN has a very useful summary of the sentence, the losses in the resolution and the related parts of the source words can be made up.

In other words, the pilot signals in LSTM and attention mechanism are important for the function of the CNN-based encoder, which can be seen from the difference between BLEU values obtained by CNN and LSSVM. CNN + LSTM can further benefit from the dependency structure encoded in the source language in the input. The dependency initial can be used to further improve the CNN + LSTM model. In LSTM, a marker bit (0 or 1) is added to the words embedded in the input layer as a marker whether they belong to the source word or not. In order to merge the dependency header information, we extend the tagging rule by adding another marker bit (0 or 1) to the original LSTM word embedding to indicate whether it is part of the dependency header of the adjunct. For example, if x_i is the embedding of the related source word and x_j is the dependency header of word x_i, the extended input of LSTM will contain as the following equation:

$$\begin{cases} x_i^{(\text{AFF,NON–HEAD})} = \left[x_i^T,\ 1,\ 0\right]^T, \\ x_j^{(\text{NON–AFF,HEAD})} = \left[x_j^T, 0, 1\right]^T. \end{cases} \tag{20}$$

The second part of the experiment is to cluster the sentences in the corpus according to the length and then test the translation model with different length sentences to verify the ability of the translation model. The test results are shown in Table 2. It can be seen from Table 2 that the performance of the proposed model is still the best in the case of different sentence length, which proves that this model also has strong translation performance in the aspect of long sentence translation. Moreover, with the increase of sentence length, the decline of the translation performance of this model is smaller than that of the traditional model, which indicates that it is less sensitive to the change of sentence length and more accurate.

Table 1: Comparison of the method with published studies.

Model	MT04/BLEU	MT05/BLEU	Average/BLEU
LSTM	15.62	14.81	15.21
LSSVM	17.76	16.89	17.32
CNN	13.12	14.32	13.72
CNN + LSTM	18.99	17.03	18.01
LSTM + ATT	21.69	23.44	22.57
Proposed method (CNN + LSTM + ATT)	24.33	25.97	25.15

Table 2: Comparison of the method with published studies.

Model	MT04/BLEU	MT05/BLEU	Average/BLEU
LSTM	13.31	12.51	12.91
LSSVM	15.21	14.35	14.78
CNN	11.06	11.53	11.3
CNN + LSTM	17.22	14.97	16.10
LSTM + ATT	20.01	21.23	20.62
Proposed method (CNN + LSTM + ATT)	22.16	24.83	23.50

5. Conclusions

Aiming at the problems of inaccurate translation and incomplete semantics in the traditional encoder-decoder translation model, we propose a hybrid neural network that combines CNN and LSTM and introduce the attention mechanism. This model can improve the performance of the context semantic connection and parallel operation and then effectively improve the translation quality of long sentences. The experimental results show that the translation performance of the improved hybrid neural network translation model is significantly better than that of the traditional translation model. In the long sentence translation test, this model also has the best performance.

In future, we will study the new English translation scenario based on the large-scale regions. In addition, an English translation system platform will be implemented.

Data Availability

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

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] L. Lemao, T. Watanabe, E. Sumita et al., "Additive neural networks for statistical machine translation," in *Proceedings of the International Conference on Parallel & Distributed Systems*, Seoul, Korea, December 1997.

[2] D. Xiong, Z. Min, and H. Li, "Enhancing language models in statistical machine translation with backward N-grams and mutual information triggers," in *Proceedings of the Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, OR, USA, June 2011.

[3] P. Li, Y. Liu, and M. Sun, "Recursive autoencoders for ITG-based translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, NJ, USA, October 2013.

[4] L. Deng, M. Seltzer, D. Yu et al., "Binary coding of speech spectrograms using a deep auto-encoder," *Interspeech*, 2010.

[5] M. Junczys-Dowmunt, T. Dwojak, and H. Hoang, "Is neural machine translation ready for deployment? A case study on 30 translation directions," in *Proceedings of the Ninth International Workshop on Spoken Language Translation (IWSLT)*, Seattle, WA, USA, October 2016.

[6] R. Sennrich, B. Haddow, and A. Birch, "Edinburgh neural machine translation systems for WMT 16," in *Proceedings of the First Conference on Machine Translation*, Berlin, Germany, August 2016.

[7] Y. Wu, M. Schuster, Z. Chen et al., "Google's neural machine translation system: bridging the gap between human and machine translation," 2016, https://arxiv.org/abs/1609.08144l.

[8] Z. Jie, C. Ying, X. Wang et al., "Deep recurrent models with fast-forward connections for neural machine translatin," *Transactions of the Association for Computational Linguistics*, vol. 4, no. 2, 2016.

[9] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, NJ, USA, October 2013.

[10] J. Brea, W. Senn, and J. P. Pfister, "Sequence learning with hidden units in spiking neural networks," in *Proceedings of the Part of Advances in Neural Information Processing Systems 24 (NIPS 2011)*, Granada, Spain, December 2011.

[11] Z. Huang, M. Cmejrek, and B. Zhou, "Soft syntactic constraints for hierarchical phrase-based translation using latent syntactic distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, USA, October 2010.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, Montreal, Canada, December 2014.

[13] K. Cho, B. V. Merrienboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[14] L. Bentivogli, A. Bisazza, M. Cettolo et al., "Neural versus phrase-based machine translation quality: a case study," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, November 2016.

[15] Z. Tu, Z. Lu, L. Yang et al., "Modeling coverage for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August 2016.

[16] G. Druck, K. Ganchev, and J. V Graça, "Rich prior knowledge in learning for natural language processing," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, Portland, OR, USA, June 2011.

[17] Z. Tu, Y. Liu, L. Shang et al., "Neural machine translation with reconstruction," 2016, https://arxiv.org/abs/1611.01874.

[18] C. Yong, X. Wei, Z. He et al., "Semi-supervised learning for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August 2016.

[19] C. T. Chung, C. Y. Tsai, H. H. Lu et al., "An iterative deep learning framework for unsupervised discovery of speech features and linguistic units with applications on spoken term detection," 2016, https://arxiv.org/abs/1602.00426.

[20] K. Xu, J. Ba, R. Kiros et al., "Show, attend and tell: neural image caption generation with visual attention," *Computer Science*, vol. 37, pp. 2048–2057, 2015.

[21] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, https://arxiv.org/abs/1508.04025.

[22] L. Liu, M. Utiyama, A. Finch et al., "Neural machine translation with supervised attention," in *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan, December 2016.

[23] S. Maskey and B. Zhou, "Unsupervised deep belief features for speech translation," *Interspeech*, 2012.

[24] L. Hao, S. Liang, J. Ye, and Z. Xu, "TensorD: a tensor decomposition library in TensorFlow," *Neurocomputing*, vol. 318, pp. 196–200, 2018.

[25] Y. Bo, R. Liu, and D. He, "Research on remote sensing image classification based on improved decision tree classification algorithm," *Computer Measurement & Control*, vol. 41, 2018.

[26] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 6, no. 99, pp. 1662–1669, 2018.

[27] Y. Zhai, X. Ding, X. Jin et al., "Adaptive LSSVM based iterative prediction method for NOx concentration prediction in coal-fired power plant considering system delay," *Applied Soft Computing*, vol. 89, 2020.

[28] X. Zhou, Y. Yang, H. E. University et al., "Study on evaluation of tense accuracy in CNN-based google translation from English to Chinese," *Modern Electronics Technique*, 2019.

[29] S. Gundapu and R. Mamidi, "Multichannel LSTM-CNN for Telugu technical domain identification," 2021, https://arxiv.org/abs/2102.12179.

[30] Y. Wang, M. Huang, X. Zhu et al., "Attention-based LSTM for aspect-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, November 2016.