ITC 3/53

Information Technology and Control
Vol. 53 / No. 3/ 2024
pp. 943-959
DOI 10.5755/j01.itc.53.3.34619

Research on Intelligent Translation Method for Short Texts Based on Improved RNN Algorithm

Received 2023/07/12 | Accepted after revision 2023/12/24

HOW TO CITE: Wang, Y., Wang, Y. (2024). Research on Intelligent Translation Method for Short Texts Based on Improved RNN Algorithm. *Information Technology and Control,* 53(3), 943-959. https://doi.org/10.5755/j01.itc.53.3.34619

Research on Intelligent Translation Method for Short Texts Based on Improved RNN Algorithm

Yan Wang

Chinese Language and Literature Department, Northwest Minzu University, Lanzhou 730030, Gansu, China
School of Foreign Languages, Lanzhou University of Arts and Science, Lanzhou 730000, Gansu, China

Ying Wang

Foreigner Language Department, Lanzhou Technology and Business College, Lanzhou, 730101, Gansu, China

Corresponding authors: Yan_Wang2023@outlook.com

As the trend towards internationalization accelerates and communication between countries and peoples becomes more important, the need for language translation becomes more urgent. Machine translation has received much attention as it is more labor and material efficient than human translation. However, current machine translation is still far from being fully automated and of high quality. The CRNN-embed model uses characters as input to the translation model, and proposes a word vector generation method with embedded CRNN, namely CRNN-embed. The model adopts a bidirectional GRU structure and introduces two attention mechanisms, CA-Cross Att and MC-SefAtt. The BLEU value of the CRNN-embed model improved by 2.57 percentage points compared to the baseline system after the attention mechanism was introduced. The BLEU values of the study model were higher than both the RNN-search and RNN-embed models, by 0.43 percentage points and 0.96 percentage points in char1, 2.02 percentage points and 3.06 percentage points in char2, respectively. As the size of the dataset increased, the model's BLEU values and n-word accuracy also increased, and its translations improved significantly. The accuracy and fluency of this model are higher than those of the traditional neural machine translation model. The study model had better translation results and was superior among similar translation models.

KEYWORDS: Improved RNN; Short text; Intelligent translation; Attention mechanism; Encoder-decoder.

1. Introduction

As globalization accelerates and countries and nations become more connected, the need for accurate translation between languages has become a pressing one [26]. Although professional human translators can accurately convey the meaning of a language, the human cost is relatively high and time-consuming, so machine translation is becoming more and more popular [14]. However, the adoption of machine translation has not been fully automated and of high quality. Current algorithms such as deep learning are often used in machine translation and have made breakthroughs, but neural machine translation mainly uses words as its basic input unit, which can make the process cumbersome and cause problems such as translation errors [18, 27]. In addition, when the lexicon is too large, word-based translation models can also cause problems such as the high dimensionality of the network model and difficulties in processing unregistered words [30]. Therefore, this paper further improves the translation model based on neural network by using character-level bilingual data as input. Under the overall framework of encoder-decoder, the neural machine translation model is further improved, focusing on strengthening the ability of character feature expression. At the same time, the traditional attention mechanism only pays attention to the corresponding relationship between the current output and the overall input, and enhances the global weight value by integrating historical information to further strengthen the word alignment effect. The aim of the research is to improve the translation effect and the efficiency of machine translation by improving the expression model. This research mainly includes five parts. In the first part of this paper, the research background and significance of machine translation are briefly introduced. The current problems of neural machine translation and the relevant solutions are presented. The content of the second part is a summary of machine translation, which mainly introduces the previous research results. This paper summarizes and analyzes the design difficulties and shortcomings of machine translation model. The third part is the research method content, mainly divided into two sections. In Section 3.1, an improved RNN algorithm-based word vector generation and language model module design is proposed. In Section 3.2, a short text intelligent translation model based on improved word vector generation and language model module is designed. The fourth part is the validity verification of the research model. The fifth part is the summary of the most research methods and the analysis of the experimental results. At the same time, the shortcomings of research methods and the direction of future research are put forward.

2. Related Work

In recent years, all sectors of society have shown a great deal of interest in machine translation and have shown strong support for it in practical terms. At the same time, as the language connection between nations becomes closer, the accurate translation and expression of language have become an urgent need of people. In this environment, the search for more effective machine translation algorithms continues to have important academic and industrial applications. Munz et al. proposed an improved machine translation system, which is based on visualised neural machine translation. And the results showed that the system could effectively improve the translation quality [17]. Garcia et al. compared and analyzed Transformers and Recurrent Neural Networks (RNNs), two kinds of attention architectures. Using several models that combine this architecture, two parallel corpora and two tokenization techniques, a neural machine translation (NMT) based on Nahuatl is proposed. The results show that the model has good translation performance [8]. For Non-Autoregressive neural machine translation (NAT) output cannot be correctly evaluated due to multimodal problems, they proposed a model that uses sequence-level training targets to train the NAT model. The results showed that the model has some superiority [23]. Satir and Bulut proposed a hybrid system-based approach, and the experimental results showed the applicability and effectiveness of the approach [22]. Lee et al. proposed an attentional mechanism translation model based on reinforcement learning, which solves the translation problem of insufficient delay in online scenarios. And the results show that the model can achieve a satisfactory translation effect [11].

Li et al. proposed a method to improve neural machine translation using latent feature feedback, which was experimentally shown to significantly outperform a strong baseline with or without denoising autoencoder pre-training [13]. Turganbayeva and Tukeyev proposed a training model to solve the unknown word search dictionary in neural machine translation. And the experiments show the effectiveness of the method [24]. Armengol-Estapé and Costa-Jussà proposed a machine translation model that introduces any number of word features into the source sequence of the attention system, and the results show the model has some advantages [1]. Marie Benjamin et al. propose a method to address the deterioration of translation quality when translating noisy texts in order to address the deterioration of translation quality when translating noisy text, Benjamin et al. proposed a neural machine translation system with different but complementary synthetic parallel data, and the results showed that the system was well suited to the problem [4]. Jian et al. proposed an attention-embedded Long Short-Term Memory (LSTM) English the results show that the research model can improve the performance and translation quality of English machine translation models [9].

The research on neural machine translation by scholars at home and abroad shows that the current neural machine translation technology is still immature, and there are still many problems that need to be studied more deeply. Neural machine translation is based on neural network, and the outstanding problem of low interpretability is difficult to avoid at this stage.

Moreover, the effect of neural network is closely related to whether the parameter Settings are reasonable, which has not yet been perfected. The effect of neural machine translation has not yet met the requirements of full automatic and high quality, and there is a large room for improvement. Therefore, based on the previous research, the study proposes an intelligent translation method for short texts based on an improved Recurrent Neural Network (RNN) algorithm, in order to obtain a more efficient machine translation algorithm model.

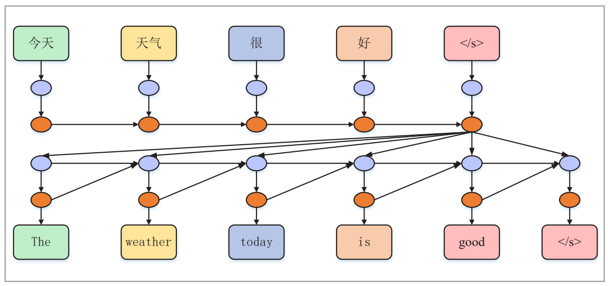
3. Design of Neural Machine Translation Model Based on Improved RNN Algorithm

3.1. Word Vector Generation and Language Model Module Design Based on Improved RNN Algorithm

End-to-end neural machine translation enables conversion from source to target language sequences, completely eliminating the need for a manually designed translation process, by building up a federated neural network to perform natural language processing tasks in an encoder-decoder framework [20]. The linear model of statistical machine translation is replaced by its nonlinear model, eliminating the need for a hidden structural pipeline and becoming a single complex neural network. The encoder-decoder framework is shown in Figure 1.

Figure 1

Encoder-Decoder framework

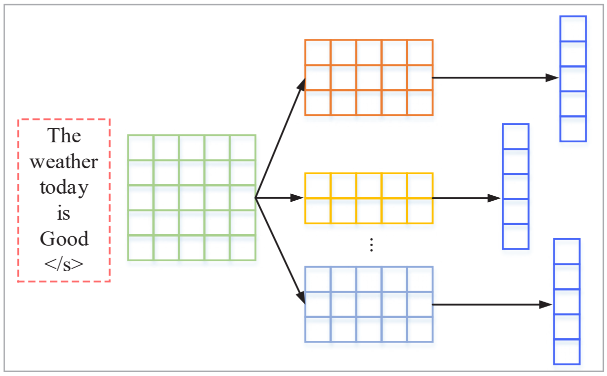


As shown in Figure 1, when a Chinese source statement is input, the model will first generate a word vector for each word. Then through the time series network model, the sentence vectors are generated in sequence. $</s>$ Representing sentence terminators, the input in the source language side generates a dense continuous representation of the sentence vector corresponding to a portion of the encoder. Target language segment corresponds to the meta-language segment, which is similar to the encoder side structure, both using a family of RNN networks that capture remote dependencies, but the encoder side is designed to generate translation outputs. As the model is temporal in nature, the output at this moment is very dependent on the output at the last moment.

Before model training, it is necessary to convert the corpus into a regular form that meets the model input requirements. For irregular corpus, a pre-processing process is required. The pre-processing of corpus mainly includes three points, which are word segmentation, named entity recognition and special vocabulary processing. Among them, the quality of participle directly affects the subsequent processing process. Compared with Chinese word segmentation, English word segmentation is much simpler. Regular expressions are used to divide words according to space symbols, and high-frequency words need to be filtered out. Then through the stem extraction process, the different deformation words are merged. The combination of Chinese characters is relatively deterministic, and the probability or frequency of Chinese characters combination can better reflect the possibility of forming words. By counting the co-occurrence frequency or probability of combinations of Chinese characters, the word segmentation method based on statistics finds out the combinations whose probability or frequency is greater than the set threshold value and determines these as words. After the preprocessing is completed, the structured bilingual data can be obtained.

In natural language processing, one-hot is one of the most widely used and straightforward representations and is known as one-hot coding. However, based on the fact that one-hot word vectors suffer from problems such as dimensional catastrophe, word embedding methods that can over-solve the problems in one-hot representations have been proposed. Usually, the training of word embeddings is done together with the training of modules such as language models, where the word vectors are obtained while the language model is being trained. Word vector length is considered to be custom, independent of dictionaries, generally 100, 200, 300 at maximum. In natural language processing, the input is usually a whole sentence in the form of a matrix and a document. Each row of the matrix represents a vector representation of a word or character, often in the form of a word embedding. The method of word2vec is adopted in the research, and all word vectors are randomly initialized, which needs to be continuously optimized in the training process. A sentence in the input layer can be likened to a two-dimensional image, where each row is a word representation vector. Vertical is the sequential representation of each word of the sentence. The size of the input data can be compared to the size of the image and can be represented by nxk. Where n represents the number of words or characters in the longest sentence in the training data. If the sentence is less than n words or characters, the zero method is adopted. k is the dimension of embedding. In the experiment, it is set to 128. In natural language processing problems, since a whole row of inputs represents a word representation vector, the width of the filter is equal to the dimensionality of its input, i.e. the word representation vector, and the length of the filter is the next number of times that can be covered by the convolution operation. Figure 2 shows the convolution process of word vectors [31].

Figure 2
Convolution diagram of word vectors



As in Figure 2, although the size of the filter is variable, usually the window size does not exceed 5 lines. When using a filter of length not 1 to do convolution in the edge region, there are no top and bottom elements adjacent to the actual and end elements. Therefore, the study performs convolution for each word with filters of different sizes by complementing the elements at that position with 0 and setting the positions where the filter has no elements to 0. In the convolution process, each convolution kernel is moved in steps of one, and multiple outputs are obtained by convolving the input matrix, which is combined to obtain a feature matrix that is the input to the next step of the recurrent neural network, which enhances the representational properties of individual characters. If a phrase occurs at one place in the sentence, the output value of the filter at that place will be very large, and the output value at other places will be very small. This preserves information about whether a feature is present in the sentence. The word vector corresponding to the last character of each word should be the word representation of the whole word, and the representation of each remaining character should be the zero vector. Although the input is processed by a layer of convolutional neural network, the convolutional neural network is only a further abstraction of the input features. The research uses RNN-embed, a recurrent neural network model for word vector generation, which incorporates a word cut switch and zeroes out the hidden layer information at the appropriate time to disconnect the preceding and following characters and generate independent word vectors. In addition, the research compensates for information prior to the word vector being input to this language model. By compensating for the information, and the blocking gradient backflow caused by the zero vector in part of the language model can be avoided, which is calculated as in Equation (1) [19].

$$\begin{cases} e_{i-1}^{x} = (1-w_i)\overline{U_c}x_i + w_i \circ g_{i-1} \\ g_{end} = (1-w_i) \end{cases} . \tag{1}$$

In Equation (1), x_i represents the input character at the time of i, g_i represents the hidden layer neuron of the recurrent neural network, and $\overrightarrow{U_c}$ represents any vector. w_i represents the switch that controls word slicing, which determines how close the input at is to the input at i $i-1$. No matter what kind of translation model is used, decoding is an essential step in the translation process. Decoding is to traverse the set space of solutions under the premise of source language sentences and parameter models to find the most reasonable translated sentences and take them as the final translation result of the model. Traversal is a search process. Heuristic depth-first search can be used to decode traversal. The search behavior depends on the calculation results of different modules, and these modules refer to these results to decide the direction of the next step until the translation terminator is generated, then the search process ends. To ensure that a meaningful vector is generated as a last resort, the word vector corresponding to w_i is forced to be identified as g_{end}. e_{i-1}^{x} The result of the decoder-side word vector module is represented, which is constrained by the word cut switch at the time of i. The language model can be understood as a model identifying the probability of occurrence of an utterance, i.e. denoted as $P(W_1, W_2, W_3, \ldots, W_K)$. If words are used as the basic units of a sentence, the probability value of a sentence S consisting of words $W_1, W_2, W_3, \ldots, W_K$ in the sequence is calculated as in Equation (2) [21].

$$P(S) = P(W_1, W_2, W_3, \ldots, W_K) . \tag{2}$$

If only the bi-gram model is considered, $P(W_K | W_1, W_2, W_3, \ldots, W_{K-1}) = P(W_K | W_{K-1})$, then Equation (2) can be equated with Equation (3).

$$P(S) = P(W_1)P(W_2|W_1)P(W_3|W_2)\ldots P(W_k|W_{k-1}) \tag{3}$$

The computation of Equation (3) is usually designed with a large number of parameters, so simpler and more efficient methods are needed, such as neural networks, etc. The RNN model ensures that the goal of the implementation is to propagate the content information through multiple iterations of the steps. This is because in the backward propagation stage, the gradient contribution value gradually decreases in the initial step propagation, and the information of long sentences will be diluted as the content increases. Because RNNs are difficult to optimize, Gated Recurrent unit (GRU) are used to replace RNNS. Because it has a more durable memory and can support longer sequences, the model is also simpler and easier to train than LSTM. The language model is generated from the current moment input and the top-clad moment hidden information state to generate the current moment hidden state and thus the output, which involves a reset gate, an update gate, and an output gate, respectively, and also includes a new memory unit. The language model is calculated as in Equation (4) [28].

$$\begin{cases} r_i = \sigma\left(W_r e_i^x + U_i h_{i-1}\right) \\ z_i = \sigma\left(W_z e_i^x + U_z h_{i-1}\right) \\ \overrightarrow{h_i} = \tanh\left(W e_i^x + U\left[R_i \circ h_{i-1}\right]\right) \\ h_i = (1-z_i) \circ h_{i-1} + z_i \circ \overrightarrow{h_i} \\ s_i = \sigma\left(W_s e_i^x + U_s s_{i-1}\right) \circ \tanh(h_i) \end{cases} \tag{4}$$

In Equation (4), σ is the sigmoid, which is used to taper the output to [0,1]. e_i^x, z_i, r_i, s_i denote the word vector, update gate, reset gate, and output gate, respectively. $W_z, U_z, W_r, U_r, W_s, U_s, W, U$ In the decoder, the input of the language model is the output of the word vector generating module, and the overall bi-directional GRU is used with an additional output gate to generate the final language output sequence with combined forward and reverse outputs. The calculation formula of the language model is shown in Equation (5) [25].

$$\begin{cases} r_i = \sigma\left(W_r e_{i-1}^y + U_r s_{i-1}^y + C_r c_i\right) \\ z_i = \sigma\left(W_z e_{i-1}^y + U_z s_{i-1}^y + C_z c_i\right) \\ h_i^y = \tanh\left(W_s e_{i-1}^y + U_s\left[r_i \circ s_{i-1}^y\right] + C_z c_i\right) \\ s_i^y = \left(1 - z_i\right) \circ s_{i-1}^y + z_i \circ h_i^y \end{cases} \tag{5}$$

In Equation (5), the dimension of is W_z, W_r, W_s $n*m$, the dimension of U_z, U_r, U_s is $n*n$ and the dimension of C_z, C_r, C_s is $n*n$. The results generated by the s_i^y language model is mainly due to the fact that the last output module on the decoder side is output in temporal order, using a bi-directional GRU structure. The information before each reference cannot be predicted at the current moment for the temporal information that follows. In natural language, word vectors and language models play an important role. In addition, they are also part of the encoder and decoder framework.

3.1. Design of an Intelligent Translation Model for Short Texts Based on Improved Word Vector Generation and Language Modeling Modules

Numerous microtransaction machines have encoders and decoders. The encoder corresponds to the source language, while the decoder corresponds to the target language. Word alignment means establishing a link between the two ends of the origin and target language, learning the latter two correspondences from a parallel corpus, learning translation rules based on this, and then completing the training of a neural machine translation model [10]. The usual representation of word alignment is $i \rightarrow j$, that is, the target word at position i corresponds to the source language word at position j. Chinese and English word order is very similar, can achieve a certain degree of matching, but cannot fully meet the matching requirements. Therefore, fuzzy mechanism can be used to match the whole. The calculation method of fuzzy matching degree of any two words $C1$ and $C2$ is shown in the Formula (6).

$$\mathrm{si}\,m\left(C1, C2\right) = \frac{2 * \left(C1 \cap C2\right)}{\left|C1\right| + \left|C2\right|}. \tag{6}$$

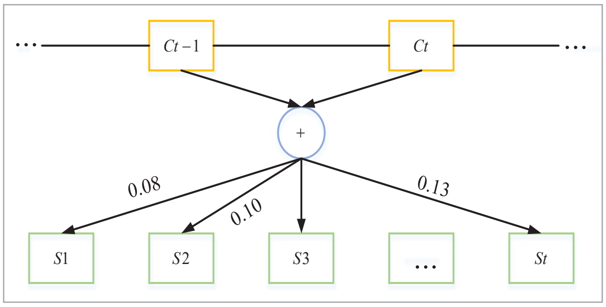
In Formula (6), $\left|C1\right| + \left|C2\right|$ Represents the number of words that intersect the two words. $\left|C1\right|, \left|C2\right|$ Corresponding to the total number of words each of these two words. If one of the two words is English e and the other is Chinese C, the calculation method is as shown in formula (7).

$$\begin{cases} DTSim\left(e, c\right) = \max_{d \in DTe}\left(Sim\left(d, c\right)\right) + q \\ q = \left(Count_{d \in DTe}\left(Sim\left(d, c\right) > h_1\right) - 1\right) \times 0.1 \end{cases} \tag{7}$$

In Equation (7), $Count()$ represents the frequency statistical function, DTe represents all translations of e, and h_1 is the threshold of similarity. The study provides performance improvements to the translation maze model by augmenting the small poly component-based sentence representation with learning knowledge of the source-side clause alignment. The decoder is translated by relying only on the hidden state of the encoder. The last vector which must encode all the content of the source sentence, i.e. the word embedding. Most translations are benchmarked with English-like word order sentences such as French, but some languages have sentences where the last word is highly linguistic in the first word of the English translation, and backwards input would make the results worse, so attention mechanisms are proposed to solve such problems. Two attention mechanisms were used, MC-SefAtt, a multiplexed collaborative self-attention mechanism, and CA-CrossAtt, a clause alignment attention mechanism [12, 16].

From Figure 3, where S_t represents the input state, and C_t represents the decoder output word. The language model uses a bidirectional loop network where the words output by each decoder depend on a weighted combination of all the input states. The result of weight summation is usually normalized to 1. The weight determines how much each input state

Figure 3
Attention mechanism



contributes to the output state. When the weight is large, the decoder will pay more attention to the corresponding part of the sentence in the original text when generating the current word of the translation. The study uses an attention mechanism that can escape the limitations of the fixed vector, with the decoder incorporating a different part of the original text for each output word generated. Most importantly the study lets the model decide on the association with the current output based on the input sentences and what has been generated. Between languages that are very similar in form, the decoder chooses to engage things sequentially and so on. A two-way recurrent network is used into the language model, where the current word output from each decoder is determined by a combination of weights from all input states, and the weights are what determine the size of each input state's contribution to the output state. The weight is usually calculated from the language model output of the source language and the hidden layer information of the target language at a time. In addition, it should be related to the historical information of the weight. The calculation method is shown in Formula (8).

$$\begin{cases} e_{ij} = V_a^T \tanh\left(W_a s_{i-1}^y + U_a s_j^x\right) \\ b_{ij}^h = \sum_{i=1}^{t-1} \exp\left(e_{ij}\right) \\ b_{ij} = \dfrac{\exp\left(e_{ij}\right)}{b_{ij}^h} \\ \alpha_{ij} = \dfrac{Tx}{j=1} \alpha_{ij} s_j \end{cases} \tag{8}$$

In Formula (8), W_a, U_a, V_a^T represents the weight matrix. s_{i-1}^y is the output of the language model on the target language side at the previous moment, and s_j^x is the output of the language model on the source language side. In the attention calculation process, the magnitude of the correlation needs to be calculated for each source language word as well as for each target language word, and the attention value needs to be calculated separately for each input-output combination. The attention mechanism simply gives the network model access to its internal memory, i.e. the hidden state of the encoder. The memory access mechanism here is a weighted combination of the multiple memory locations retrieved by the model, which has the advantage of allowing for easier end-to-end network model training via a back-propagation algorithm. To obtain the translation results, e_{i-1}^x, c_i and s_i^y are incorporated at each moment befo zr e the terminators are generated. The output module is calculated as in Equation (9).

$$t_i = \sigma\left(W_i e_{i-1}^y + U_i s_i^y + C_i c_i\right). \tag{9}$$

In Equation (9), W_i, U_i, C_i all represent the weight matrix with dimensions of $2l*m, 2l*n, 2l*n^{`}$, respectively. The generated probability values for the output module are shown in Equation (10).

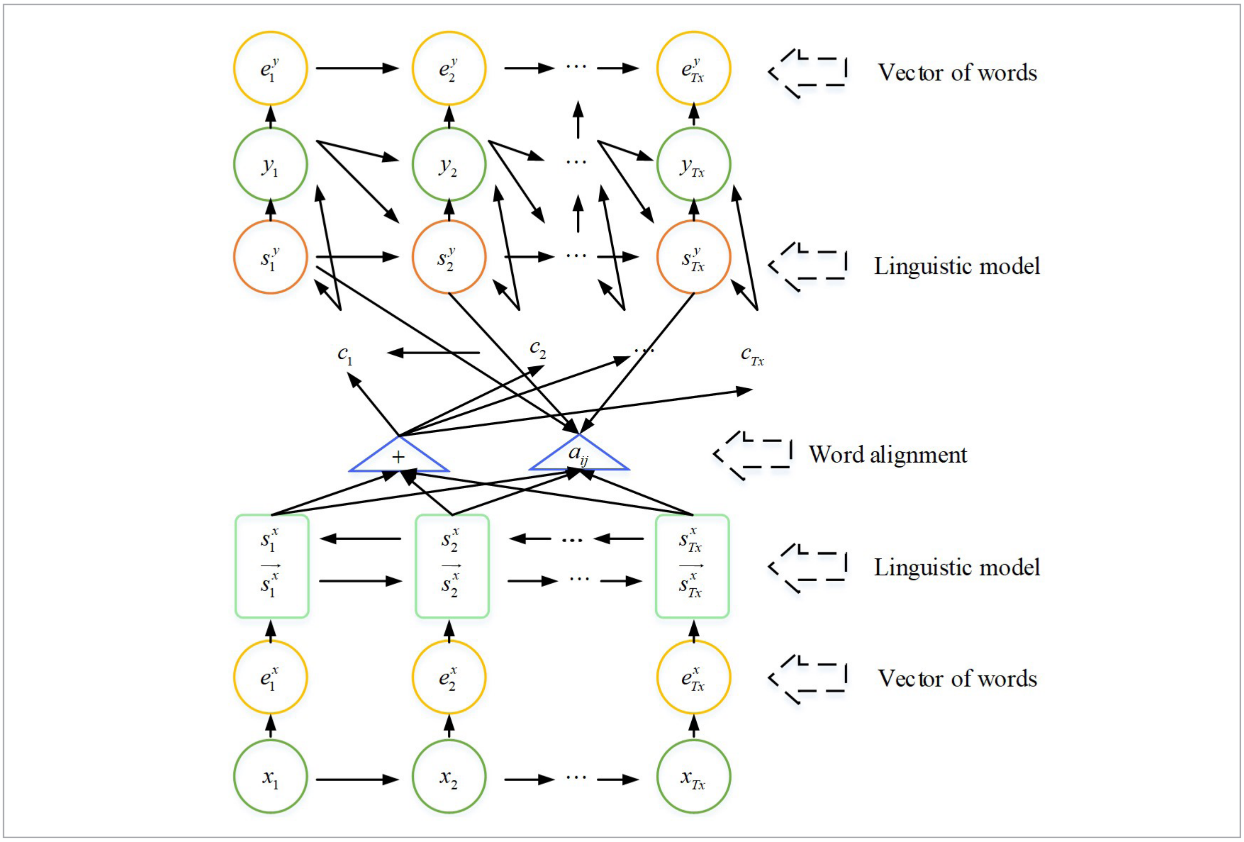
$$\begin{cases} t_i = \max\left\{t_{i,2j-1}, t_{i,2j}\right\}, j = 1, \ldots, l \\ p\left(y_i \middle| e_{i=1}^y, s_i^y, c_i\right) = y_i^T soft\max\left(V_p t_i\right) \end{cases}. \tag{10}$$

In Equation (10), V_p represents the weight matrix with dimensions of $k_y * l$, respectively. max represents the non-linear activation function, and *soft*max represents the activation function. The output module and the word vector generation work together with the language model in order to form the complete decoder side, thus completing the complete translation process. Figure 4 shows the overall frame structure of the model.

As shown in Figure 4, the entire response process of the model is implemented in the encoder-decoder framework. Starting from the decoder side, the word vector generation module is input into the source language sequence and the related word vector. The encoder language model sequence is then generated by the language module, and the word vector and language model are calculated in two steps from the forward and reverse directions, finally synthesising

Figure 4

Overall framework of translation model



the language model output on the forward and reverse source language side. The word alignment module acts as a bridge between the encoder and the decoder, firstly budgeting the correlation between each output step and the individual inputs on the encoder side and performing a weighted sum, and then carrying out calculations on the decoder side to generate the translated output. The decoder side calculates the language model and the translation result at the current moment based on the output of all previous calculations. The final translation output is then used to generate the cloud side of the word vector model in preparation for the subsequent output. The size of the batch has an important impact on gradient descent methods in machine learning. When the error of a single sample is calculated each time and the gradient correction is carried out, a mutually cancelling effect may be formed due to the difference of each sample. As a result, the result oscillates back and forth and cannot be convergent. Therefore, using the batch method to select the appropriate size, not only can effectively improve the utilization of memory, but also speed up the processing speed. However, if you blindly increase the batch size, you may exceed the range of memory can afford. Therefore, you need to choose a moderate value, such as 80. In this research, the study uses a batch approach, which will be set to 80, and each gradient to be updated is calculated as in Equation (11) [15].

$$x_{i+1} = x_t + \Delta x_t \,. \tag{11}$$

In Equation (11), x_t represents the parameter to be updated and t represents the moment. Each gradient update is calculated as in Equation (12).

$$\Delta x_t = -\eta \cdot g_t \,. \tag{12}$$

In Equation (9), g_t represents the gradient and η represents the learning rate. Changing the learning rate during each update can speed up the convergence, and Adadelta's method was used in this study. $\Delta\theta_t$ The parameters to be updated are calculated as in Equation (13) [5].

$$\Delta\theta_t = \frac{RMS\left[\Delta\theta\right]_{t-1}}{RMS\left[g\right]_t} g_t . \quad (13)$$

In Equation (13), RMS denotes the root mean square and $\Delta\theta_t$ denotes the parameter to be updated. g_t The rescaled gradient performs the update calculation as in Equation (14).

$$x_t = x_{t-1} - g_t^{'} . \quad (14)$$

In Equation (14), x_t denotes the leaked mean used to store the second order inverse of the change in the parameters of the model itself. $g_t^{'}$ denotes the adjusted gradient, which is calculated as in Equation (15) [7].

$$g_t^{'} = \frac{\sqrt{\Delta x_{t-1} + \varepsilon}}{\sqrt{s_t + \varepsilon}} \odot g_t . \quad (15)$$

In Equation (15), s_t represents the average of the leaks used for the second order derivatives and Δx_{t-1} represents the average of the leaks from $g_t^{'}$. The Adadelta algorithm is calculated as in Equation (16).

$$E\left(g^2\right) = \rho\Delta x_{t-1} + \left(1-\rho\right)\bullet\left(g_t^{'}\right)^2 . \quad (16)$$

In Equation (16), ρ represents the parameter du jour. The study uses $BLEU$ as the evaluation method for this translation model [2]. For a sentence with a translation, the candidate translation is represented by C, the corresponding reference English by S and the set of word-length phrases by N-gram. P_n As a precision measure, the study introduced a penalty factor (Brevity Penalty, BP), as in Equation (17), since it cannot evaluate the completeness of the translation [29].

$$\mathrm{BP} = \begin{cases} 1 & if \quad c > r \\ \mathrm{e}^{(1-r/c)} & if \quad c > r \end{cases} . \quad (17)$$

In Equation (17), c stands for candidate translation length, and r the effective length of the reference translation. $BLEU$ This is actually a pooled weighted average of the accuracy of N-gram and is calculated as in Equation (18).

$$BLEU = BP \times \exp\left(\sum_{n=1}^{N} w_N \times \log P_n\right) \quad (18)$$

In Equation (18), $N = 1,2,3,4$, w_n denotes the corresponding weight of the contributing n meta-word, which is generally taken as a constant value, i.e. $1/n$.

4. Experimental Results Analysis of Short Text Intelligent Translation Model Based on Improved RNN Algorithm

The theano-based deep learning framework is used for the experiments, as it consists of a large number of convenient optimisation libraries that can be used as the basis for the multidimensional arrays of universities and make full use of the GRU structure, thus greatly improving the computational efficiency. In order to achieve good training results, an early stopping mechanism is proposed. The operating system used for the experiments is 64-bit Ubuntu 14.04 with 64G of RAM and a 2.5TB hard disk. In this study, bilingual oral corpus pairs in Chinese and English are used as data sets. Since the existing fragmented data sets are far from meeting the requirements of deep learning on the size of data sets, it is necessary to obtain them by itself. The resources of individual subtitle groups are more authoritative in the same type of websites, the resources are updated relatively timely, and the data scale is larger. Therefore, the experiment takes subtitle website as the resource source of the dataset. Get tens of millions of data sets from subtitle websites through web crawlers and optimize extraction. To compare the effect of the model on word and character level input and to verify the effect of dataset size on the effect of the model, two datasets were extracted during the experimental phase, a small-scale dataset containing a bilingual training set of 1.8 million and a large-scale dataset containing a bilingual dataset of 21 million, both with a test data size of 5000. To prove that the sequence length has a significant effect on the translation effect, the sentence length at the word level and the character level in the size and scale data set is calculated respectively. As shown in Table 1.

As can be seen from the statistical results in Table 1, the sentence length of word-level Chinese sentences in the dataset is generally less than 20, and the sentence length of English sentences is generally less than 60,

Table 1
Word level million and ten million data sentence length statistics

Sentence length	Word level million data sentence length statistics				Word level ten million data sentence length statistics			
	Training Set (Chinese)	Training set (English)	Test Set (English)	Test set (English)	Training Set (Chinese)	Training set (English)	Test Set (English)	Test set (English)
1~9	1442446	127 1044	152763	142378	17647172	18753154	137397	174160
10~19	357330	523695	47038	56639	4286910	3189486	61263	25838
20~29	223	4909	197	569	9136	755	1332	2
30~39	1	349	2	360	169	7	8	0
40~49	0	3	0	51	11	3	0	0
50~59	0	0	0	3	5	0	0	0
60+	/	/	/	/	3	1	0	0

and the sentence length of bilingual pairs is shorter, which is also consistent with the characteristics of oral language. The results of comparing this model with other translation models are shown in Table 2.

Table 2 shows the BLUE values of the different network-structured translation models on the development and test sets for the Chinese-English translation task. Among the translation models without the attention mechanism, the model with the GRU structure outperforms the RNN and LSTM-only models in all tests, but is weaker than the study model with the bidirectional GRU structure, and weaker than all other translation models with the attention mechanism. Since the traditional attention mechanism only focuses on the current output and the overall input information, the global information-based word alignment method was used in this study, as shown in Figure 5. The performance improvement results of the study model with the addition of each mechanism.

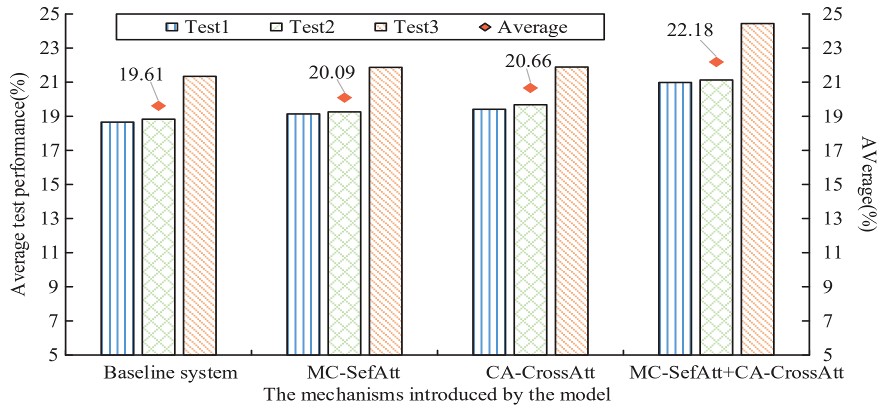
As shown in Figure 5, the study model improved its BLEU value by 0.48 percentage points over the baseline system after introducing the MC-SefAtt attention mechanism on the baseline system. The model improved its BLEU value by 1.05 percentage points over the baseline system after introducing the CA-CrossAtt attention mechanism on the baseline system. In contrast, after the model introduced both the CA-CrossAtt and MC-SefAtt attention mechanisms

Table 2
Performance comparison of translation models with or without attention mechanism

Translation model	Dev.	Test.1	Test.2	Test.3
RNN	12.49	11.97	11.73	11.89
LSTM	14.57	14.25	13.93	14.09
GRU	14.59	14.26	13.95	14.11
BiGRU	16.31	15.92	15.59	15.68
RNN+Attention	21.44	16.85	16.51	16.72
LSTM+Attention	23.37	22.84	22.57	22.63
GRU+Attention	23.41	22.87	22.59	22.65
CRNN-embed	25.38	24.96	24.47	24.51

Figure 5

Performance results of the research model after introducing various mechanisms



on the baseline system, its BLEU value improved by 2.57 percentage points over the baseline system. The experimental results show that the study's introduction of two attention mechanisms incorporating the alignment knowledge approach enables the model to learn semantic structural alignment features at the bilingual clause level as well as to better improve the accuracy of the neural machine translation model. To further analyse the reasons for the improved system performance due to the study's encoders, the experiments grouped the test set by source-end sentence length and tested the BLRU scores for each group separately, the results of which are shown in Figure 6. As shown in Figure 6, the performance of this study's translation system and the baseline system are approximately the same when the sentences are short, and their performance is improved accordingly when the sentence length becomes longer. The baseline system encoder is weaker than the encoder of the study system in terms of long-time capture capability, and this fusion encoder design can effectively cover the feature extraction capability of the encoder, further verifying its effectiveness in improving the performance of the translation system. In addition, experiments were conducted to statistically analyse the results of recursively updating the source-side an-

Figure 6

The influence of the input sentence length on the performance of the fusion encoder translation system

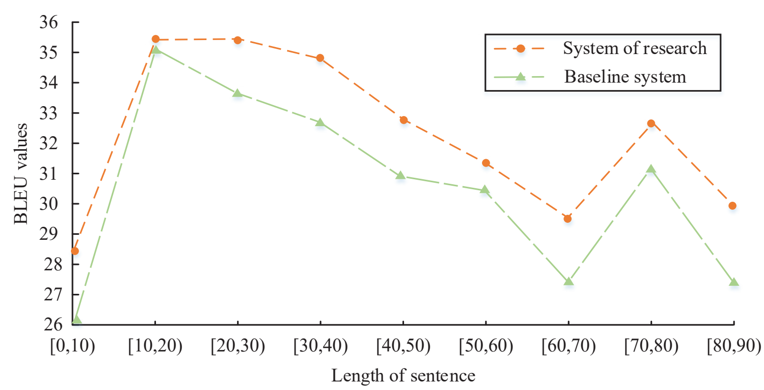
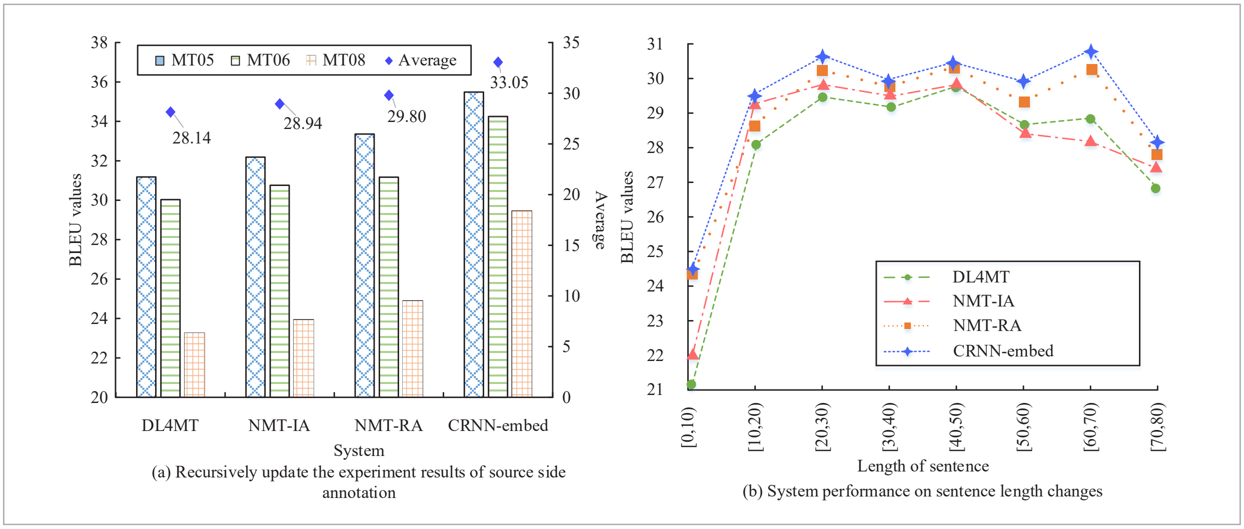


Figure 7

Performance comparison results of translation systems in the recursively updated source-side annotation experiment



(a) Recursively update the experiment results of source side annotation

(b) System performance on sentence length changes

notation of this study's improved RNN-based translation system with three other better-performing attention-based neural machine translation systems, as shown in Figure 7(a). Moreover, to further analyse and compare the performance of the four systems, the experiments grouped the test sets by source-end sentence length and tested their BLUE scores for each group separately, the results of which are shown in Figure 7(b).

As can be seen from Figure 7(a), the study's proposed translation system performs better than the other three systems in terms of the average BLEU value of the study's improved RNN-based CRNN-embed model, both when compared with the traditional attention-based system DL4MT and the better NMT-IA and NMT-RA systems, with the average BLEU value of the study's improved RNN-based CRNN-embed model being higher than that of DL4MT, NMT-IA, and NMT-RA by 4.91 percentage points, 4.11 percentage points, and 3.25 percentage points, respectively. This also indicates that the research method has improved the over- and under-translation problems of the translation system, thus enhancing the performance of the overall translation system. As can be seen in Figure 7(b), the study system achieves consistent performance improvements with the other three systems in terms of length groupings, especially for long sentences. This also indicates that the studied CRNN-embed model is more effective for long-term memory and can better fuse information from the source and target ends, thus achieving an improvement in system performance. Table 3 below shows the translation results of different models, GNMT for the word and character-based neural machine translation model introduced by Google, the RNN-search model, the RNN-embed model, and the study's proposed CRNN-embed model.

In Table 3, 6 indicators are corresponding to the different calculations of the evaluation method. $BLEU$ For the main evaluation metrics, BP is determined by the length, and $P1$ to $P4$ represent the accuracy rates for one-to-four words, respectively. There, No. 1 and No. 2 are the evaluation results for small and large quantitative sets, respectively, and it is clear from the data in the table that the performance of each model improves significantly when the size of the dataset grows. The CRNN-embed model in this study has higher values than both the RNN-search and RNN-embed models, with 0.43 and 0.96 percentage points higher in char1, and 2.02 and 3.06 percentage points higher in char2, respectively. And the CRNN-embed models from $P1$ to $P4$ are higher than all the models in the table. The experiments show that the research translation models translate better and are superior to similar translation models. To further demonstrate the effect of sentence length on the clas-

Table 3

Comparison of model effects

Model	*BLEU*	*P1*(%)	*P2*(%)	*P3*(%)	*P1*(%)	*BP*
GNMT	23.39	41.8	23.9	21.2	13.9	1.0
RNN-search 1	19.01	38.6	23.2	15.1	11.9	1.0
RNN-search 2	21.73	41.8	26.2	17.5	11.9	1.0
RNN-embed 1	18.48	39.6	23.2	14.3	9.2	1.0
RNN-embed 2	20.69	41.3	25.3	16.5	10.9	1.0
CRNN-embed 1	19.44	39.3	23.6	15.4	10.2	1.0
CRNN-embed 2	23.75	44.0	30.3	22.3	15.5	1.0

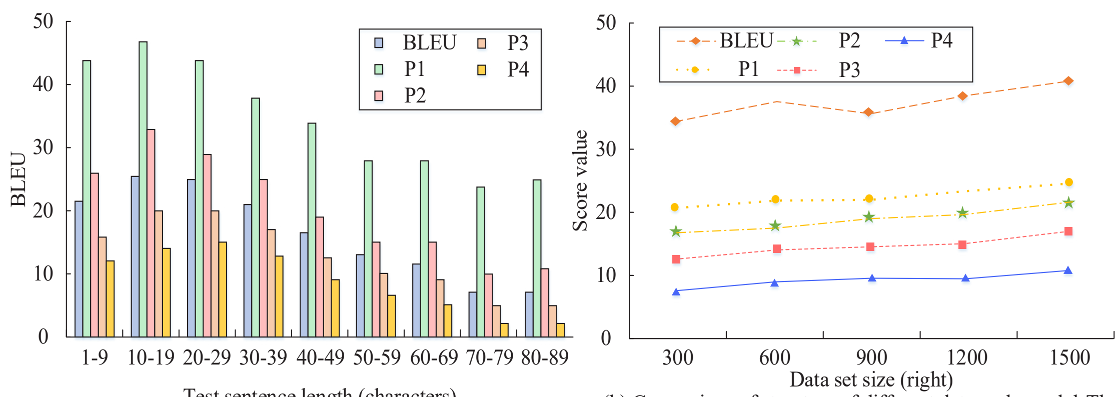
sification effect, the experiments were tested by scoring the models by sentence length, and the results are shown in Figure 8(a). In addition, to verify the effect of dataset size growth on model effectiveness, the experiments were trained with data sets of 3 million, 6 million, 9 million, 12 million and 15 million, and the results are shown in Figure 8(b).

As can be seen in Figure 8(a), the model scores are higher for lengths below 20, and decrease as the sentence length increases. From Figure 8(b), it can be seen that as the size of the dataset increases, the model's BLEU values and n-word accuracy rates also increase and its translations improve significantly. The experiments were concluded with a manual evaluation of the translation method, using two dimensions, accuracy and fluency, to evaluate the research CRNN-embed model against the traditional neural machine translation model, and the results of their scores are shown in Figure 9.

From Figure 9, the CRNN-embed model designed in this study has a higher manual score in terms of translation accuracy and fluency than the traditional neural machine translation model, with an average score of 93.84, 4.99 points higher than the traditional model. And the root-mean-square error of CRNN-embed model is only 0.14, which is 0.18 lower than the traditional model. The error rate of the research model is lower than that of the traditional method, and the translation effect is better. In addition to translating English sentences, between lan-

Figure 8

Comparison of different data scale models and classification results by sentence length

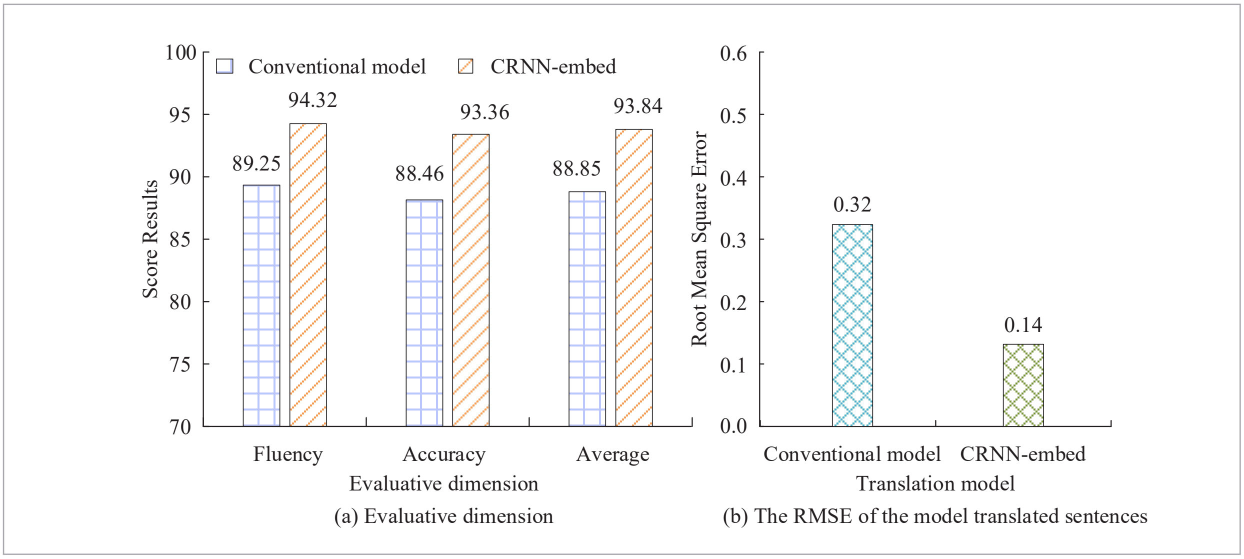


(a) The results were classified by word length

(b) Comparison of structure of different data scale modelsThe results were classified by word length

Figure 9

Comparison results of model translation effects



guages with very similar forms, such as Chinese and Korean, the decoder may choose to participate in things sequentially, participating in the first word of the original when the first Korean word is generated, and so on. In order to further verify the translation performance of the research model, The experiment will study the translation effect of the model on different languages and several advanced translation models, such as attention mechanism translation model based on reinforcement learning [11], improved neural machine translation algorithm based on latent feature feedback [13], neural machine translation algorithm of clinical text between distant languages [29], and improved language translation algorithm based on hidden Markov model [6]. The translation results of each model are shown in Table 4.

As can be seen from Table 4, firstly, in Korean data sets, the studied CRNN-embed model processes more sentences per second than other models. At the same time, the research model also processes more words per second than other models. In the Serbian data set, the CRNN-embed model also processes more sentences and words per second than the other models.

Table 4

Comparison of translation effects of different language concentration models

Model	BLEU	Sentences processed per second		Words processed per second		Accuracy rate (%)
		Korean data set	Serbian language data set	Korean data set	Serbian language data set	
Baseline system	43.33	42.46	32.23	873.99	765.82	78.6
References [11]	41.23	45.12	34.45	922.65	843.84	83.1
References [13]	43.15	47.46	38.42	1108.06	956.51	84.3
References [29]	43.87	47.34	38.62	1109.42	956.21	86.8
References [6]	44.12	47.48	38.53	1112.33	956.54	87.4
CRNN-embed	48.47	58.76	51.76	1209.59	1174.32	92.1

In addition, the average translation accuracy of the research model is also the highest, at 92.1%. In conclusion, the translation performance of the research model is better in different language datasets. The classical attention mechanism only focuses on the corresponding relationship between the current output and the overall input, while the translation model designed in this study is based on the overall encoder-decoder framework, which emphasizes the enhancement of character feature expression ability. By integrating historical information, this study further strengthens the word alignment effect, thus obtaining better short text translation effect.

5. Conclusion

As the trend towards internationalisation accelerates, communication between countries and peoples becomes more closely important, and the resulting need for language translation becomes more urgent. This research proposes an improved RNN algorithm for the intelligent translation of short texts, which uses characters as input to a neural machine translation model, and improves components such as word vector generation for changes in the form of the input. The results show that the study model outperforms other models without attention mechanisms or GRU structures in terms of performance, with a BLUE value of 25.38. The model improves its BLEU value by 2.57 percentage points over the baseline system after introducing both CA-CrossAtt and MC-SefAtt attention mechanisms on the baseline system. The mean performance BLEU values of the studied CRNN-embed model were 4.91 percentage points, 4.11 percentage points, and 3.25 percentage points higher than DL4MT, NMT-IA, and NMT-RA, respectively. The BLEU values of the study model were higher than both the RNN-search and RNN-embed models, by 0.43 percentage points and 0.96 percentage points in char1, and 2.02 percentage points and 3.06 percentage points in char2, respectively, providing superiority among similar translation models. As the size of the dataset increased, the model's BLEU values and n-word accuracy also increased, and its translation results were significantly improved. The manual ratings of the study model were higher than those of the traditional neural machine translation model in terms of both translation accuracy and fluency, indicating that the study model was more effective in translation. Although the study has comprehensively validated the effectiveness and superiority of the translation model, there is still room for improvement and the model structure will continue to be optimised in subsequent studies in the hope of further reducing the complexity of its training.

Funding

The research is supported by: the National Social Science Research Fund of 2021--A Study on the Cultural Memory and Translation of the Painted Pottery Culture Along the Yellow River: The Cultural Terms Collection and its Bilingual Dictionary Compilation (No. 21BYY067); 2022 Higher Education Innovation Fund Project of Gansu Province: Research on the Application of English Translation and Bilingual Parallel Corpus for Gansu Red Tourism Publicity (No. 2022A-175).

References

1. Armengol-Estapé, J., Costa-Jussà, M. R. Semantic and Syntactic Information for Neural Machine Translation. Machine Translation, 2021, 35, 3-17. https://doi.org/10.1007/s10590-021-09264-2
2. Batanović, V., Cvetanović, M., Nikolić, B. Fine-Grained Semantic Textual Similarity for Serbian. Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), 2018.
3. Batanović, V., Cvetanović, M., Nikolić, B. A Versatile Framework for Resource-Limited Sentiment Articulation, Annotation, and Analysis of Short Texts. PLoS One, 2020, 15(11), e0242050. https://doi.org/10.1371/journal.pone.0242050
4. Benjamin, M., Atsushi, F. Synthesizing Parallel Data of User-Generated Texts with Zero-Shot Neural Machine Translation. Transactions of the Association for Computational Linguistics, 2020, 8, 710-725. https://doi.org/10.1162/tacl_a_00341
5. Berrichi, S., Mazroui, A. A Word Alignment Study to Improve the Reliability of the Statistical and Neural Translation System. International Journal of Net-

working and Virtual Organisations, 2022, 26, 104-124. https://doi.org/10.1504/IJNVO.2022.10045915

6. Chang, Y., Wang, X., Xue, M., Liu, Y., Jiang, F. Improving Language Translation Using the Hidden Markov Model. Computers, Materials and Continuum, 2021, 67(3), 3921-3931. https://doi.org/10.32604/cmc.2021.012304
7. Dedes, K., Utama, A. B. P., Wibawa, A. P., Afandi, A. N., Handayani, A. N., Hernandez, L. Neural Machine Translation of Spanish-English Food Recipes Using LSTM. JOIV: International Journal on Informatics Visualization, 2022, 6(2), 290-297. https://doi.org/10.30630/joiv.6.2.804
8. García, S., Lucero E Huerta, E., Hernández, J., Cruz, J., Méndez, B. Implementation of Neural Machine Translation for Nahuatl as a Web Platform: A Focus on Text Programming and Computer Software, 2021, 47(8), 778-792. https://doi.org/10.1134/S0361768821080168
9. Jian, L., Xiang, H., Le, G. LSTM-Based Attentional Embedding for English Machine Translation. Scientific Programming, 2022, 10, 3-8. https://doi.org/10.1155/2022/3909726
10. Komorniczak, J., Zyblewski, P., Ksieniewicz, P. Statistical Drift Detection Ensemble for Batch Processing of Data Streams. Knowledge-Based Systems, 2022, 252(Sep.27), 1-18. https://doi.org/10.1016/j.knosys.2022.109380
11. Lee, Y. H., Shin, J. H., Kim, Y. K. Simultaneous Neural Machine Translation with a Reinforced Attention Mechanism. ETRI Journal, 2021, 43(5), 775-786. https://doi.org/10.4218/etrij.2020-0358
12. Li, W., Qi, F., Tang, M., Yu, Z. Bidirectional LSTM with Self-Attention Mechanism and Multi-Channel Features for Sentiment Classification. Neurocomputing, 2020, 387(Apr.28), 63-77. https://doi.org/10.1016/j.neucom.2020.01.006
13. Li, Y., Li, J., Zhang, M. Improving Neural Machine Translation with Latent Features Feedback. Neurocomputing, 2021, 463, 368-378. https://doi.org/10.1016/j.neucom.2021.08.019
14. Li, Y., Zhang, G., Wang, Y. Research on Reliability Allocation Technology for NC Machine Tool Meta-Action. Quality and Reliability Engineering International, 2019, 35(6), 2016-2044. https://doi.org/10.1002/qre.2489
15. Lin, C. Application of Traditional Cultural Symbols in Art Design under the Background of Artificial Intelligence. Hindawi Limited, 2021, 2021(Pt.44), 1-11. https://doi.org/10.1155/2021/1258080
16. Liu, R., Ning, X., Cai, W., Li, G. Multiscale Dense Cross-Attention Mechanism with Covariance Pooling for Hyperspectral Image Scene Classification. Hindawi, 2021, 2021(Pt.3), 1-15. https://doi.org/10.1155/2021/9962057
17. Munz, T., Väth, D., Kuznecov, P., Vu, N. T., Weiskopf, D. Visualization-Based Improvement of Neural Machine Translation. Computers & Graphics, 2022, 103, 45-60. https://doi.org/10.1016/j.cag.2021.12.003
18. Peris, A., Casacuberta, F. Online Learning for Effort Reduction in Interactive Neural Machine Translation. Computer Speech & Language, 2018, 58, 98-126. https://doi.org/10.1016/j.csl.2019.04.001
19. Pozharkova, I., Antamoshkin, O. The Method of Context-Dependent Annotation Based on the Spectral Language Model in Information Systems for Monitoring Emergencies. 2021, 229, 714-721. https://doi.org/10.1007/978-3-030-77445-5_64
20. Ren, Q., Su, Y., Wu, N. Research on Mongolian-Chinese Machine Translation Based on the End-to-End Neural Network. International Journal of Wavelets Multiresolution & Information Processing, 2020, 18(1), 46-59.
21. Ronghui, L., Xinhong, W. Application of Improved Convolutional Neural Network in Text Classification. IAENG International Journal of Computer Science, 2022, 49(3), 762-767.
22. Satir, E., Bulut, H. Preventing Translation Quality Deterioration Caused by Beam Search Decoding in Neural Machine Translation Using Information Sciences, 2021, 581, 791-807. https://doi.org/10.1016/j.ins.2021.10.006
23. Shao, C., Feng, Y., Zhang, J., Meng, F., Zhou, J. Sequence-Level Training for Non-Autoregressive Neural Machine Translation. Computational Linguistics, 2021, 47(4), 891-925. https://doi.org/10.1162/coli_a_00421
24. Turganbayeva, A., Tukeyev, U. The Solution of the Problem of Unknown Words Under Neural Machine Translation of the Kazakh Language. Journal of Information and Telecommunication, 2021, 5(2), 214-225.
25. Vanajakshi, P. G., Mathivanan, M., Kumaran, T. S. Investigation on Large Vocabulary Continuous Kannada Speech Recognition. International Journal of Biomedical Engineering and Technology, 2021, 36(1), 1-24. https://doi.org/10.1504/IJBET.2021.115984
26. Vu, K., Haraguchi, N., Amann, J. Deindustrialization in Developed Countries Amid Accelerated Globalization: Patterns, Influencers, and Policy Structural Change and Economic Dynamics, 2021, 59, 454-469. https://doi.org/10.1016/j.strueco.2021.09.013

27. Wang, F., Chen, W., Yang, Z. Hybrid Attention for Chinese Character-Level Neural Machine Translation. Neurocomputing, 2019, 35, 44-52. https://doi.org/10.1016/j.neucom.2019.05.032
28. Wang, Q., Li, C. Incident Detection and Classification in Renewable Energy News Using Pre-Trained Language Models on Deep Neural Networks. Journal of Computational Methods in Sciences and Engineering, 2022, 22(1), 57-76. https://doi.org/10.3233/JCM-215594
29. Xabier, S., Olatz, P., Gorka, L., Oronoz, M. Neural Machine Translation of Clinical Texts Between Long Distance Languages. Journal of the American Medical Informatics Association, 2019, 12, 1478-1487. https://doi.org/10.1093/jamia/ocz110
30. Xia, Y. Research on Statistical Machine Translation Model Based on Deep Neural Network. Computing, 2020, 102(3), 643-661. https://doi.org/10.1007/s00607-019-00752-1
31. Zhang, Y., Wang, Q., Li, Y. Sentiment Classification Based on Piecewise Pooling Convolutional Neural Network. Computers, Materials & Continua, 2018, 56(2), 285-297.