Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **4**, November **2025**

A Study on Chinese Semantic Role Annotation Based on Bidirectional LSTM Neural Networks

Bing-Rui Ren*

School of Chinese Language and Literature
Central China Normal University
Wuhan 430079, P. R. China
18635063025@163.com

Benjamin Thomas Johnson

Fayette Institute of Technology
Oak Hill 25901, USA
Benjamin.TJ@fayettetech.edu

*Corresponding author: Bing-Rui Ren

Received September 5, 2024, revised April 9, 2025, accepted August 12, 2025.

Abstract. *Semantic role annotation is a fundamental application of Natural Language Processing (NLP), and deep learning is mainly adopted at present, but it is tough to establish semantic links between words that are far away from each other in a long sentence, which leads to low performance on long sentences. Therefore, this paper suggests a Chinese semantic role annotation method based on bidirectional LSTM neural network (BiLSTM). To address the issue of uneven distribution of semantic role labels in the dataset, associative learning is used to select sentences similar to the target sentence and their related information from the annotated dataset, and the BERT model is introduced for encoding to obtain the output representation of associative learning. The final word representation is obtained by stitching the association learning representation, lexical representation and target word representation, and the attention mechanism is introduced to model the semantic importance of each word in the sentence. The sentence is then encoded by BiLSTM to solve the long-distance dependency problem. Finally, the optimal tag sequence is predicted by global tag optimisation using Conditional Random Field (CRF). The results on the Chinese corpus CPB imply that the suggested approach improves the F1 values on long sentence intervals by 14.27% and 5.3% compared to the other two models, and performs better in the annotation of semantic role tasks.*

Keywords: Semantic role labelling; Bidirectional LSTM; Attention mechanism; BERT model; Conditional random field

1. **Introduction.** Natural language processing (NLP) technology uses natural language to achieve effective communication and exchange between humans and machines, so that people can use computers to carry out intelligent processing of massive data. As one of the eight major languages in the world, Chinese is the language with the largest number of language applications in the world, and perfecting Chinese NLP technology is of great significance for enhancing the core competitiveness of China [1]. Chinese semantic role annotation (CSRA) is a common implementation of shallow semantic analysis, which focuses on the predicate of a sentence and does not provide in-depth analysis of the semantic information contained in the sentence, but only analyses the semantic relationship between each word in the sentence and the corresponding predicate, and makes

the corresponding semantic marking [2, 3]. CSRA, as a basic technology of NLP, is an essential part of many NLP tasks, and the good or bad annotation results will have a subtle impact on the actual completion of natural language processing tasks. Therefore, how to improve the performance of CSRA algorithms is one of the most popular research contents at present [4].

1.1. **Related work.** Semantic role annotation was first studied by Gildea et al. [5], who used syntactic tree features to identify the relationship between predicates and arguments in Chinese corpus. Early CSRA models mainly used traditional machine learning algorithms, Xue [6] used decision tree algorithm to conduct CSRA experiments, but this algorithm is very limited in dealing with high-dimensional data. Wan et al. [7] used support vector machine to achieve good results, but the efficiency is very low. In addition, most of the CSRA models based on traditional machine learning rely on syntactic analysis and feature extraction. Zhou and Xue [8] extracted features such as predicates and lexical properties from syntactic trees and combined them, and then used a maximum entropy classifier for semantic role annotation, with an F1 value of 75.60%. Wang et al. [9] fused three features of combinatorial category, phrase structure and dependent syntactic analysis, which is rich in information but noisy. Wang et al. [10] used the Chinese frame network as experimental data, and adopted the Conditional Random Field (CRF) model [11] to identify and classify semantic roles at the same time, and finally achieved 72% of the F1 value. As deep neural networks rapidly growing, there have been many studies using neural network models for semantic role annotation. Huang and Chen [12] used a multi-feature fusion neural network structure to construct a CSRA model, and the final F1 value on the CPB dataset reached 70.54%. Shen et al. [13] used convolutional neural network for CSRA, but it is easy to cause overfitting when the dataset size is too small. To address this issue, Song et al. [14] designed an RNN-based CSRA method, which exploits the long-distance information in the sequences, and the experimental outcome on a Chinese proposition corpus achieved 77.09% of F1 value. Lin et al. [15] offered a simple architecture for semantic role annotation, Deep Attentional Neural Networks, which is based on self-attention mechanisms and RNNs, and is capable of directly capturing the relationship between pre- and post-texts. Gers and Schmidhuber [16] verified that LSTM outperforms traditional RNN and demonstrated that LSTM performs better in context-independent linguistic benchmarks of RNN. Jin et al. [17] adopted an LSTM model to model the context of the current word in the sentence, and then spliced together the feature vector representations, and finally predicted the tagged categories using a CRF. Although the traditional LSTM model has made great progress compared to RNN, it cannot capture the semantics of long-distance sentences well due to its own structural defects, and bidirectional LSTM (BiLSTM) deals with this issue well. Wang et al. [18] exploited the advantages of BiLSTM network and correlation network to jointly train frame disambiguation task and semantic role labelling task, and the F1 value was improved by 5.7%. Su et al. [19] fully integrated the syntactic path information into the BiLSTM model, and experimentally proved that it could effectively improve the results.

1.2. **Contribution.** Although deep neural networks have achieved some success in CSRA, they need to be based on the difficulty of establishing semantic links between words that are far away from each other in long sentences, resulting in low performance on long sentences. Intending to the above issues, this paper designs a CSRA model for BiLSTM. Firstly, to deal with the issue of imbalanced distribution of semantic role labels, an association learning representation is used to select sentences similar to the target sentence as association sentences from the annotated dataset without introducing external resources. Then the associative learning representation, target word representation, etc. are spliced and fused to get the final word representation, and the attention mechanism is used to enhance the attention to the keywords. Then the sentences are encoded by a bidirectional LSTM encoder to solve the long- distance dependency problem in the semantic role annotation task. Finally, the semantic role annotation results are obtained after global normalization using CRF. Comparison experiments are conducted on the Chinese Anticipation CPB, and the results imply that the precision rate, recall rate and F1 value of the suggested model are 90.05%, 88.19 and 89.11%, respectively, which are better than the comparison model, and exhibit excellent performance in semantic role annotation.

2. Theoretical analysis.

2.1. LSTM network.

Traditional RNN networks cannot make use of sentence future information, and Bi-RNN can solve this defect well, but both of them cannot model long-distance information well, and it is easy to have problems such as gradient disappearance and gradient explosion, which can be solved well with the introduction of LSTM units [20, 21]. In contrast to traditional RNN networks, LSTM adds a cell state to record the information passed over time. As shown in Figure 1, LSTM uses input gates, forget gates, and output gates for information updating and utilization.

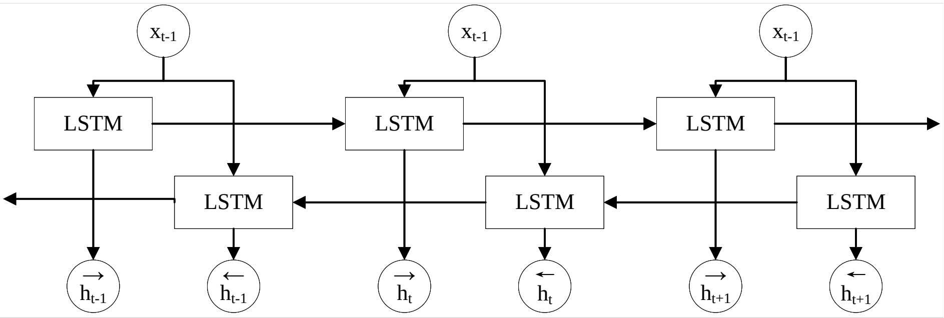
x_{t-1}
x_{t-1}
x_{t-1}
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
$\overrightarrow{h}_{t-1}$
$\overleftarrow{h}_{t-1}$
$\overrightarrow{h}_{t}$
$\overleftarrow{h}_{t}$
$\overrightarrow{h}_{t+1}$
$\overleftarrow{h}_{t+1}$

Figure 1. Schematic diagram of LSTM unit

Suppose g is the output of the LSTM unit, d is the value of the LSTM memory unit, and x is the input data. The current input data x_t, the unit output g_{t-1} at the previous time, and the unit value d_{t-1} at the previous time all affect the output at the current time.

$$\tilde{d_t} = \tanh\left(v_{xd} * x_t + v_{gd} * g_{t-1} + b_d\right) \tag{1}$$

$$i_t = \delta\left(v_{xi} * x_t + v_{gi} * g_{t-1} + v_{ci} * d_{t-1} + b_i\right) \tag{2}$$

(2) Forgetting gate: The f_t function of range $(0, 1)$ is used to control the amount of information transmitted from the cell state of d_{t-1} to the current moment d_t.

$$f_t = \delta(v_{xf} * x_t + v_{gf} * g_{t-1} + v_{cf} * d_{t-1} + b_f) \tag{3}$$

(3) Output gate: used to control the output of LSTM memory unit status value.

$$O_t = \delta(v_{xo} * x_t + v_{go} * g_{t-1} + v_{do} * d_{t-1} + b_o) \tag{4}$$

BiLSTM is an RNN composed of forward and backward LSTM, which is used to process the forward and reverse order of input sequence respectively, and can take into account all the information before and after the current moment, so as to improve the expression power and prediction accuracy. When using BiLSTM for annotation, not only the context information before each word but also the context information after each word can be considered, so that the model can capture more key information. Therefore, the BiLSTM model is used in this paper, as shown in Figure 2.

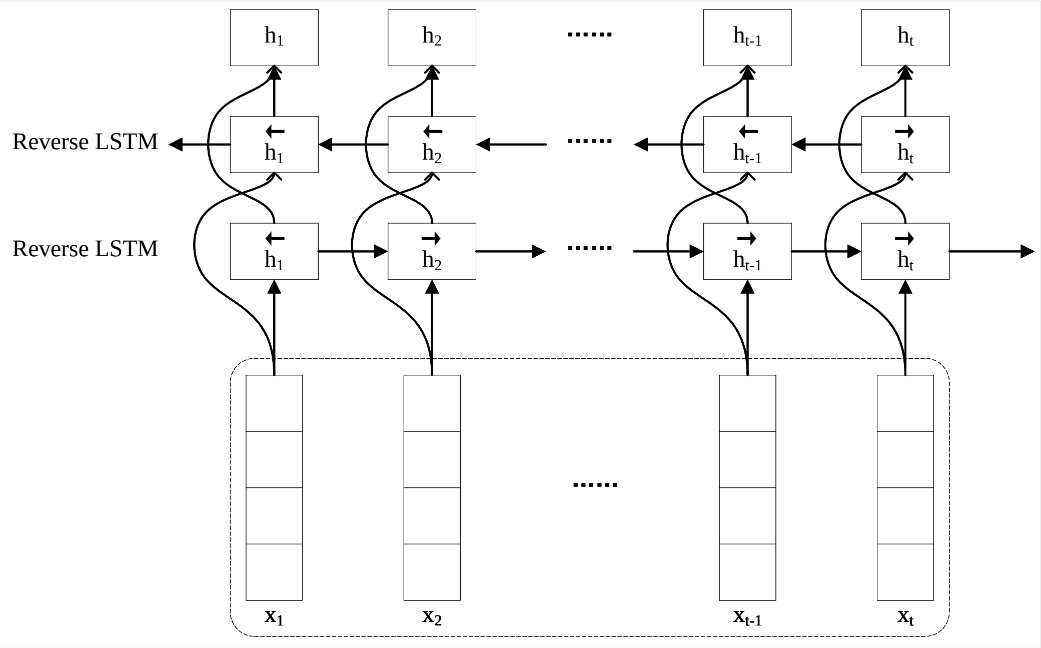

Figure 2. BiLSTM model structure diagram

2.2. **Conditional random field.** CRF combines the characteristics of maximum entropy model [22] and hidden Markov model [23], and is an undirected graph model training method that maximizes the conditional probability [24]. In the process of use, the graph structure of the linear chain is usually used for sequential sequence labeling. The linear chain component random field model defines $y = (y_1, y_2, \dots, y_i, \dots, y_n)$ for a given input sequence $x = (x_1, x_2, \dots, x_i, \dots, x_n)$ conditional probability with the parameter w to express the form of the state sequence a, which is expressed as follows.

$$P_w(y \mid x) = \frac{1}{Z_w} \exp\bigl(w^T \, \phi(x, y)\bigr) \tag{5}$$

where the feature figures $\phi(x, y) = \bigl[\phi_1 \ \phi_2 \ \Sigma \ \phi_{\text{trans}}\bigr]$, $\phi_{\text{trans}} = \bigl[c_{11}, c_{12}, \dots, c_{ij}\bigr]$ represent the observed transformation from the i to the j element in Σ, Z_w is the normalized constant with sum of 1, and the parameter estimation is solved by maximizing the trained Gaussian priori, and the calculation formula is shown as follows.

$$\sum \log\bigl(P_w(y \mid x_i)\bigr) \ - \ \sum \frac{w^2}{2\delta^2} \tag{6}$$

where δ^2 is the variance of the Gaussian segment and also the tuning parameter.

Using CRF model in CSRA can improve the accuracy of the model from many aspects. CRF can jointly model the label sequence and better capture the interaction between labels by introducing the dependency relationship between labels. Thus, CRF was used in the annotation model to improve the accuracy of the model annotation.

3. **Chinese word vector representation based on associative learning.** The complex sentence patterns of Chinese sentences make the distribution of semantic role labels in the data set very uneven, and thus the information of labeled data cannot be fully utilized in the process of model training and prediction, which affects the performance of CSRA. Therefore, based on associative learning [25], this paper selects sentences similar to the target sentences and their label information from the labeled data set to obtain the word vector representation of the target sentences, laying the foundation for subsequent CSRA. The specific process of associative learning representation is shown in Figure 3.

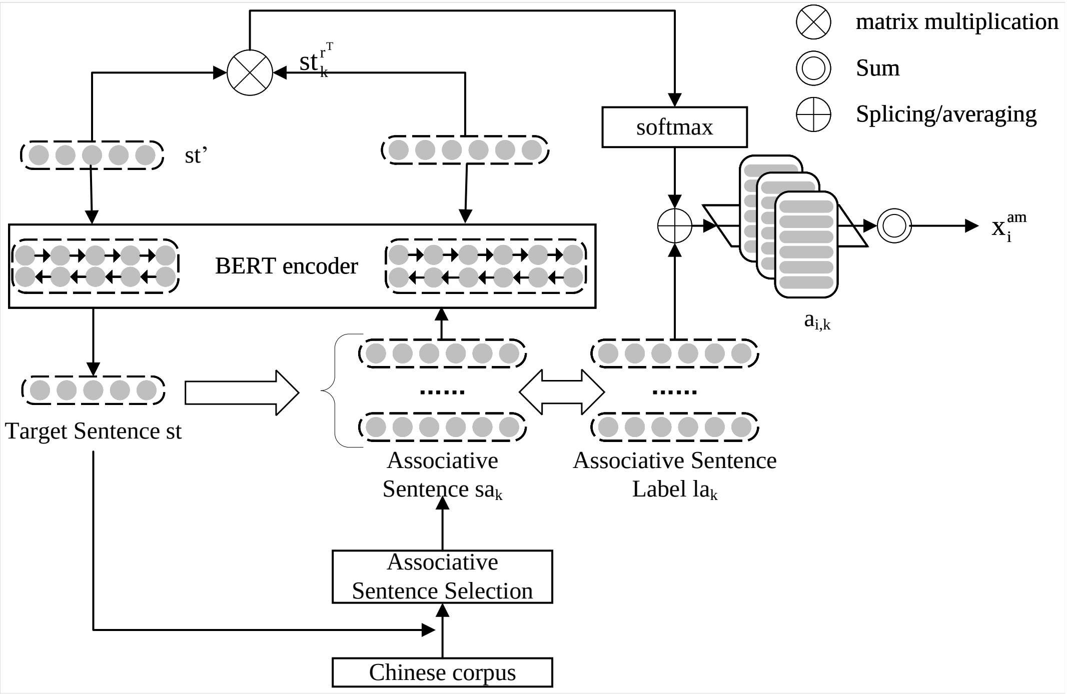

Figure 3. The specific process of associative learning representation

Associative learning is introduced in this chapter to help the model complete the CSRA for labeled sentences (target sentences) by using relevant information in labeled data. Due to the limited computing resources and the noise contained in the labeled data, the most similar and useful sentences can be selected from the labeled data set, which is called the associative sentences of the target sentences.

Suppose that the target sentence to be labeled is s_t and the sentence in the labeled data set is s_j^l, where $j \in (1, m)$ and m are the number of sentences in the data set. First, the similarity sim_j between s_t and each s_j^l is calculated. Then s_j^l is sorted according to its corresponding similarity sim_j. Finally, the first n s_j^l with the greatest similarity is selected as the associative sentence of target sentence s_t, denoted as s_k^a, and its corresponding semantic role tag sequence is l_k^a, where $k \in (1, n)$.

$$\mathrm{sim}_j = \mathrm{Similarity}(s_t, s_j^l) \tag{7}$$

The editing distance of s_t and s_j^l corresponding parts of speech sequence is taken as the similarity of two sentences. As shown in Equation (8), where POS_{s_t} represents the target sentence s_t represents the corresponding part of speech sequence, and POS_{s_j} represents the part of speech sequence corresponding to s_j^l.

$$\mathrm{Similarity}(s_t, s_j^l) = -\mathrm{Edit\_dis}(\mathrm{POS}_{s_t}, \mathrm{POS}_{s_j}) \tag{8}$$

After selecting the associative sentence of the target sentence, the target sentence and its associative sentence are transformed into the lexicon form, which are X^{s_t} and X^{s_a} respectively. The BERT encoder is used to encode s_t and s_k^a respectively, as shown in Equation (9) and Equation (10), where $k \in \{1, 2, \ldots, n\}$.

$$s_t^{\mathrm{BERT}} = \mathrm{BERT}(X^{s_t}) \tag{9}$$

$$s_{a_k}^{\mathrm{BERT}} = \mathrm{BERT}(X_k) \tag{10}$$

To make better use of the label information in associative sentences, the attention mechanism is introduced in this chapter. Firstly, for any target sentence, the similarity matrix $S = s_t s_a^T$ of s_t and each associative sentence s_k^a is calculated, and Equation (11) is used to normalize S_k for line Softmax, where $\beta_{i,k}$ represents the similarity of each word in the i-th associative sentence of s_t and its k-th associative sentence. Take $\beta_{i,k}$ as the weight coefficient to sum each element in the semantic role tag l_k^a corresponding to the k-th associative sentence of s_t, and get the Attention of the i-th word and the k-th associative sentence in s_t as

$$\beta_{i,k} = \frac{S_k(i,i')}{\sum_{i'=1}^{n_k} S_k(i,i')}, \qquad b_{i,k} = \sum_{l=1}^{n_k} \beta_{i,k}\, l_{b_k,l}. \tag{11}$$

For the goal of making better use of the label information in the associative sentence, it is merged in a representation b_i, which is merged in the following two ways.

(1) **Concatenation:** The Attention representation of the semantic role tag l_k^a corresponding to the target sentence and all the selected associative sentences is concatenated as the output representation of associative learning.

$$b_i = b_{i,1} \parallel b_{i,2} \parallel \cdots \parallel b_{i,n} \tag{12}$$

(2) **Average:** After averaging the Attention representation of the semantic role tag l_k^a corresponding to the target sentence and all the selected associative sentences, it is represented as the output of associative learning.

$$\bar{b}_i = \frac{1}{n}\sum_{k=1}^{n} b_{i,k} \tag{13}$$

Because there is some difference between the associative sentence and the target sentence, b_i contains some noise and useless information, and the gating mechanism is used to process b_i, and the final associative learning representation b_m is obtained.

$$x_i^{b_m} = \mathrm{sigmoid}(w_i^b + a_i) \tag{14}$$

4. Chinese semantic role annotation based on bidirectional LSTM neural network.

4.1. Attention representation of Chinese word feature vector.

In CSRA tasks, semantic role blocks are not always adjacent to target words, that is, there is a long-distance dependency issue, which is not considered in the existing CSRA models based on feature engineering methods. Therefore, this chapter uses the Bi-LSTM encoder to code the sentences to obtain the long distance dependency information. Meanwhile, since the semantic importance of words in sentence semantic roles is different, the attention mechanism is introduced to model the semantic importance of each word in the sentence. Finally, the labeling results are obtained by using CRF. The structure of the designed CSRA model is implied in Figure 4.

Based on the Chinese word vector association representation in the previous chapter, the proposed model generates a word for x_i for each word w_i in the sentence, where i represents the position of the word in the sentence. Each x_i is obtained by concatenating the following feature vectors: associative learning represents $x_i^{b_m}$, pre-trained word vector x_i^{pe}, randomly initialized part of speech represents x_i^{pos}, randomly initialized target word represents x_i^{target}, location represents x_i^{loc}, predicate indicator x_i^{ind}, and the final word represents $x_i = \{x_i^{b_m}, x_i^{pe}, x_i^{pos}, x_i^{\text{target}}, x_i^{\text{loc}}, x_i^{\text{ind}}\}$.

For a particular target word, the semantic importance of each word in the sentence is different. In order to make the model focus on the target words with important semantics, the final word vector needs to be represented with attention. Firstly, the word representation matrix X is nonlinearly transformed using Equation (15) to obtain the intermediate representation M. Then the similarity between the matrices M and W is computed by

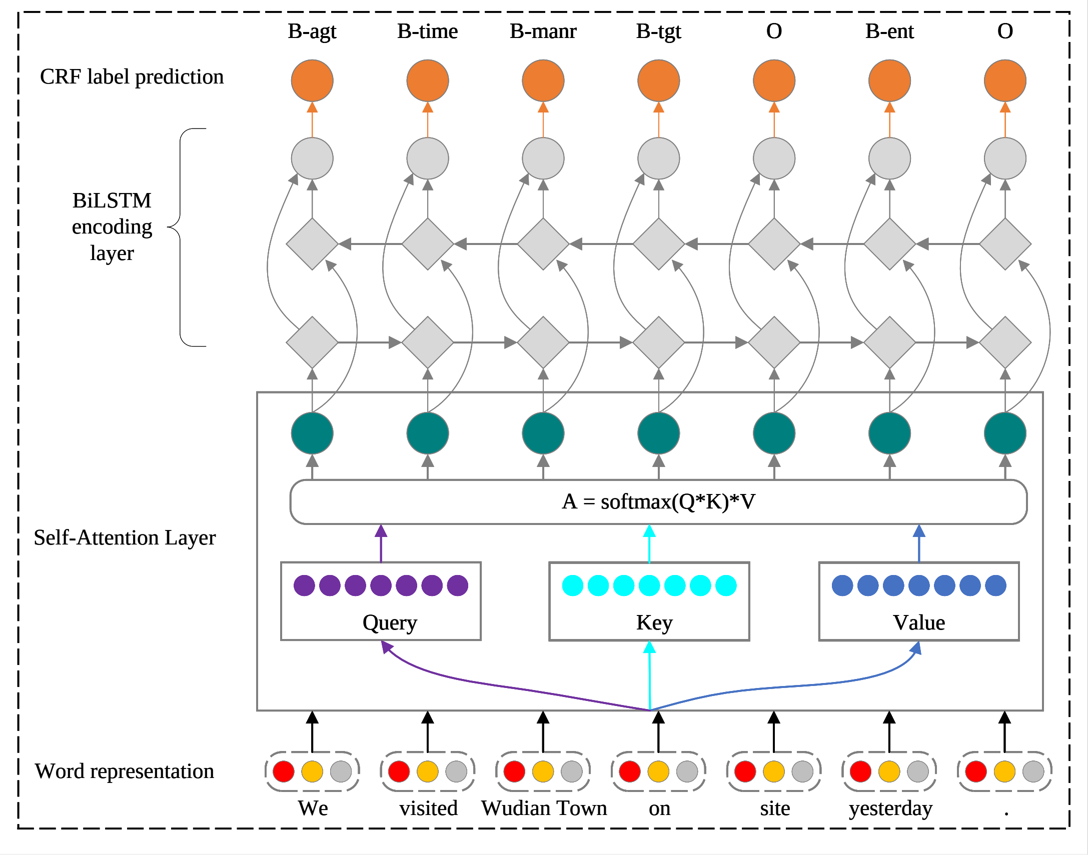

Figure 4. The framework of the designed CSRA model

Equation (16), where W is a randomly initialized trainable parameter. Equation (17) performs a normalization operation on this similarity to obtain the attention value (weight) of each word in the sentence b'. Finally, Equation (18) weights and sums the attention weights b' with the matrix X to obtain the attention representation matrix B for each word.

$$M = \tanh(X) \tag{15}$$

$$S(W, M_i) = W^T M_i \tag{16}$$

$$b'_i = \text{softmax}\big(S(W, M_i)\big) = \frac{\exp(S(W, M_i))}{\sum_j \exp(S(W, M_j))} \tag{17}$$

$$B = \sum_i b'_i X_i \tag{18}$$

4.2. **Bidirectional LSTM coding.** Since Chinese target words and semantic roles are not always adjacent to each other, the results of CSRA are poor. Therefore, in order to obtain long- distance information more effectively, a multi-level BiLSTM coding level is added after the attentional representation of Chinese sentence word vectors to obtain a richer representation and avoid the phenomenon of gradient vanishing in the coding process. In the BiLSTM network, each cell unit has $C, g_i, g_f, g_o, C_t, h_t$,where C is a candidate value for the current cell state; g is a gate controlling the flow of information; C_t is the current cell state; and h_t is the cell cryptic state, as shown in Equation (19) to Equation (22), where represents multiplication by element.

$$C = \tanh(W_c z_t + U_c h_{t-1} + a_c) \tag{19}$$

$$g_j = \text{sigmoid}(W_j z_t + U_j h_{t-1} + a_j) \tag{20}$$

$$C_t = g_i \tag{21}$$

$$h_t = g_o \tag{22}$$

The two obscured level states output from BiLSTM are spliced into a new feature fusing bi-directional information, by capturing sentence contextual semantic information from both positive and negative directions. Bidirectional information splicing is an effective means of fusing contextual information. The state of the i-th word in the sentence sequence is shown in Equation (23).

$$h_i = [h_i \oplus h_i] \tag{23}$$

where h_i denotes the hidden layer state from left to right, h_i denotes the hidden layer state from right to left, and h_i denotes the new hidden layer state with spliced context information.

Suppose the input sentence sequence is $x = (x_1, x_2, \dots, x_n)$, and the semantic feature vector of the sentence encoded by the BiLSTM network is $H = (h_1, h_2, \dots, h_n)$. The decoder computes the semantic feature vector c_t provided by the encoder at the current time t through the attention mechanism level, and Equation (24) shows the computation process of the decoder, and z_{t-1} denotes the state vector of the decoder in the hidden layer at the moment $(t-1)$.

$$e_{ti} = a(z_{t-1}, h_i) \tag{24}$$

$$c_t = \sum_{i=1}^{n} \left(\frac{\exp(e_{ti})}{\sum_{k=1}^{n} \exp(e_{tk})} h_j \right) \tag{25}$$

where e_{ti} represents the similarity score between the feature vector of the ith one in the encoder and the obscured level vector of the decoder at the time $(t-1)$, $a(.)$ represents the similarity function, c_t represents the weighted average of the output vectors of the encoder at the time t, and the label vector generated by the decoder at the time $(t-1)$ is denoted as y_{t-1}, which can be computed by the decoder's obscured state vector z_{t-1} through Equation (26).

$$z_t = f(z_{t-1}, y_{t-1}, c_t) \tag{26}$$

4.3. **Tag prediction.** Although BiLSTM can capture longer contextual information, it only considers the features of the words themselves and easily ignores the constraints between the labels and labels in label prediction, whereas CRF can learn the dependencies between the labels well and obtain the optimal sequence with the highest probability [26].

Assuming that the output sequence is $y = (y_1, y_2, \dots, y_n)$, the model predicts the score of sentence x with semantic role label y as $\text{score}(x, y)$, as shown in Equation (27). Where H denotes the state feature function computed from the output of the Bi-LSTM coding layer, and H_i denotes the score of the i-th word in sentence x labelled as y_i. S denotes the state transfer feature function, i.e., the transfer probability, and $S_{i,j}$ denotes the probability of transferring from one state y_{i-1} to another state y_i.

$$\text{score}(x, y) = \sum_i S_{y_i, y_{i-1}} + \sum_i H_i \tag{27}$$

Using Softmax function, the correct tag sequence probability value is calculated as shown in Equation (28).

$$p(y \mid x) = \tag{28}$$

$$y \in Y_x$$

where Y_x represents all the tag sequences, including possible and impossible ones, and y represents the real tag sequences. Afterwards, the log-likelihood is used to maximise the likelihood of the correct tag sequences, as shown in Equation (29).

$$p(y \mid x) = \tag{29}$$

The model is trained using the log-likelihood function [27], which corresponds to the loss function as in Equation (30).

$$\text{loss} = -\log(p \mid x) = -\log y = -\text{score}(x, y) + \log\left(\sum_{y} \exp(\text{score}(x, y))\right) \tag{30}$$

The model parameters are trained by optimizing the loss through the Stochastic Gradient Descent (SGD) optimization function [28]. As shown in Equation (31), in decoding, this chapter uses the Viterbi algorithm [29] to predict the optimal sequence of semantic role labels corresponding to the output sentence x. where Y_X denotes the sequence of all possible labels for the input sequence x.

$$y^* = \arg\max_{y \in Y_X} \text{score}(x, y) \tag{31}$$

5. Experiments and analysis of results.

5.1. Experiment on the number of model training sessions.

The text corpus used in the experimental process of this paper comes from the Chinese Corpus of the University of Pennsylvania CPB dataset [30]. There are 18418 Chinese words in the corpus, and Figure 5 shows the statistics of the proportion of different sentence types, in which the composition of sentences with more than 30 words is counted as super-long sentences; those with less than 30 words and more than 20 words are counted as long sentences; those with less than 20 words and more than 10 words are counted as medium-long sentences; and those with less than 10 words are counted as short sentences. In this paper, the CPB corpus is divided into two parts, the training set and the test set, according to the ratio of 4 : 1.

In the experiment, the dimension of word vector is 100, the dimension of features such as lexical and syntactic path features is 10, and the number of BiLSTM levels is 2. In addition, the other hyperparameters are set as the learning rate is 0.015, the discard rate *dropout* is 0.5, the number of hidden layers is 200, and the optimization function is SGD, Batch size is 80. hardware experimental platform is Inter Core i5-7500M 3.40 GHz, 8G RAM, Windows 7 system, tensorflow 1.4.0, python 3.5.2. In the performance evaluation of different models, the precision (P), recall (R) and F1 values commonly used in reviews are used and the proposed model is denoted as ABiLSTM. The comparison models are selected as LSTM-CRF model [17] and TBiLSTM model [19]. Since the experimental corpus in the input model is already tagged with labelled data, the backpropagation algorithm is used for training, in which the model learns all the training samples is the end of a training session. At the same time, it is necessary to allow the model to repeat the learning process on the training samples on a continuous basis, in order to obtain a better representation of the corpus. In the experimental process, the corpus and precomputer were handled in the same way as above. The number of training sessions (T) is set to 50, 100, 150, 200, and the results are implied in Table 1.

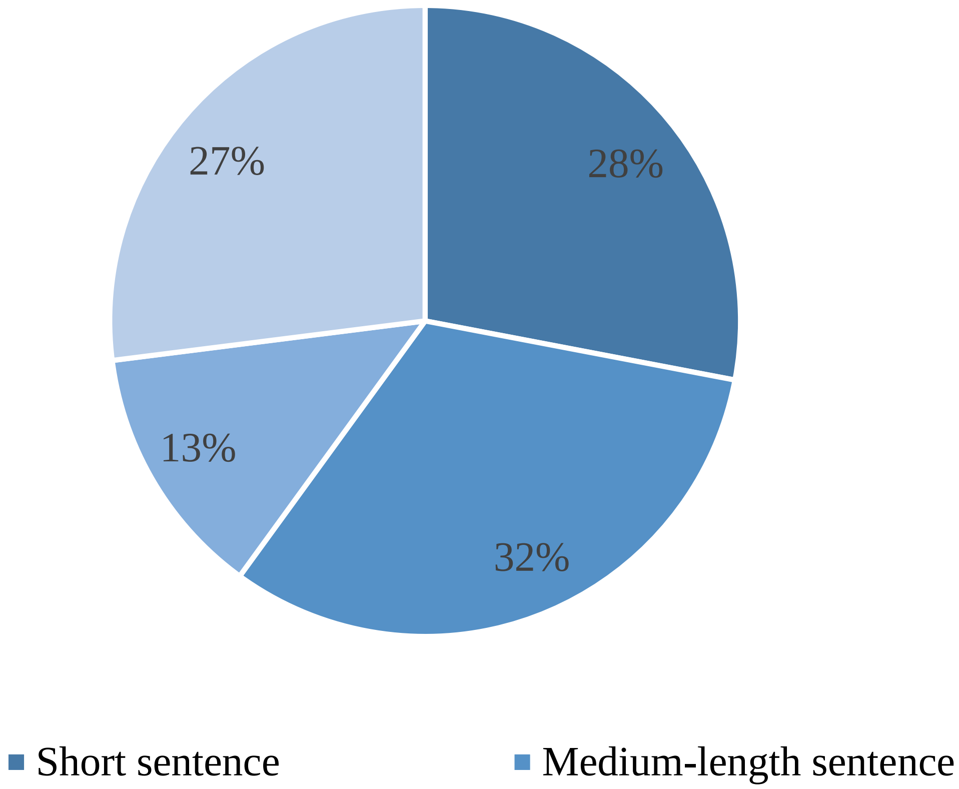

Figure 5. Statistics on the percentage of different sentence types

Table 1. Experimental results of ABiLSTM under different training times

T	LSTM-CRF			TBiLSTM			ABiLSTM		
	P/%	R/%	F1/%	P/%	R/%	F1/%	P/%	R/%	F1/%
50	73.28	74.22	73.75	81.02	78.15	79.56	85.17	83.08	84.11
100	76.26	73.95	75.09	83.71	81.29	82.48	90.05	88.19	89.11
150	81.84	79.82	80.82	86.98	85.95	86.46	87.54	84.29	85.88
200	77.41	76.94	77.17	83.96	79.82	81.84	88.12	85.66	86.87

From the experimental outcome in the above table, it can be seen that the labelling results of ABiLSTM reached the optimum when the number of training times was taken as 100, with P, R, and F1 being 90.05%, 88.19%, and 89.11%, respectively. Whereas, the LSTM-CRF and TBiLSTM annotation results reached the optimum is when the number of training is taken as 150. In the entire picture, ABiLSTM performs best in A, P, and F1 metrics under all training times, and the best F1 of ABiLSTM is improved by 8.29% and 2.65% compared to the best F1 of LSTM-CRF and TBiLSTM, respectively. LSTM-CRF models the context of a sentence based on unidirectional LSTM, but does not make effective use of the input forward and backward feature information. Although TBiLSTM fuses syntactic path information through BiLSTM, it does not make full use of the semantic role labelling information, which makes the labelled data contain noise, and thus the training results are not as good as ABiLSTM.

Different Epochs are trained with the same set of data, but the weights of the models are updated with completely different values. Because the models of different Epochs are at different positions in the cost function space, the later the model is trained, the closer to the bottom, the smaller the cost is. During the training process of the three models,

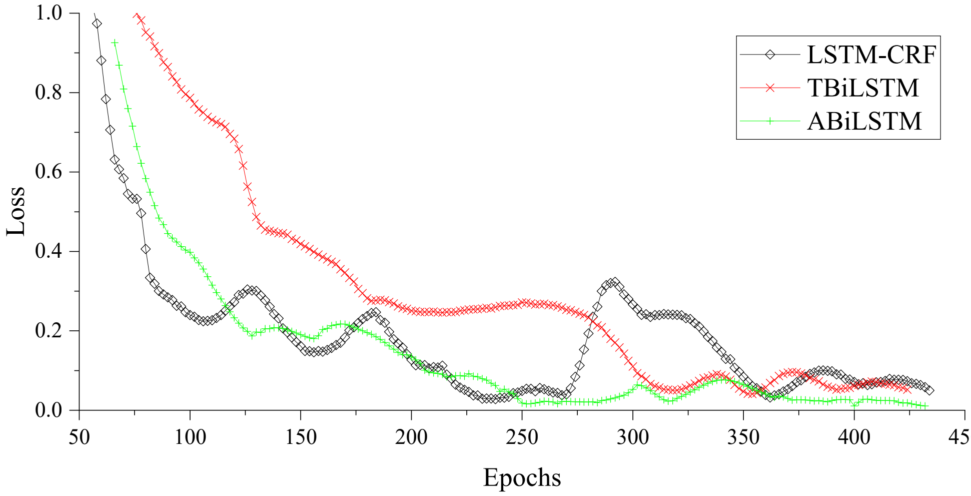

Figure 6. Comparison of losses for different models

the Loss change curves of each model were counted and integrated into one graph, and the specific results are shown in Figure 6. From Figure 6, after about 280 epochs of training, the model starts to stabilize; after about 310 epochs of training, the LSTM-CRF starts to stabilize; after about 240 epochs of training, the BiLSTM starts to stabilize; and after about 260 epochs of training, the TBiLSTM starts to stabilize. In LSTM-CRF, more data training is needed to stabilize, indicating that simple feature vector splicing makes the word vectors more irregular and reduces the original semantic expressiveness of the word vectors. As the training progresses, the change of Loss value in the late stage of TBiLSTM training is much smoother than that of other models, which indicates that the non-predicate word vectors have increased the correlation with the predicate word vectors. ABiLSTM reduces the effect of the sparse target sentences on the imbalance of the training of the model by applying the associative learning to the BiLSTM, so that the data become more representative.

5.2. **Impact of each sentence length on performance.** To compare the performance of the models on each sentence length interval, experiments were conducted on a test set of four sentence length intervals using LSTM-CRF, TBiLSTM and ABiLSTM, and the results of the experiments are shown in Figure 7.

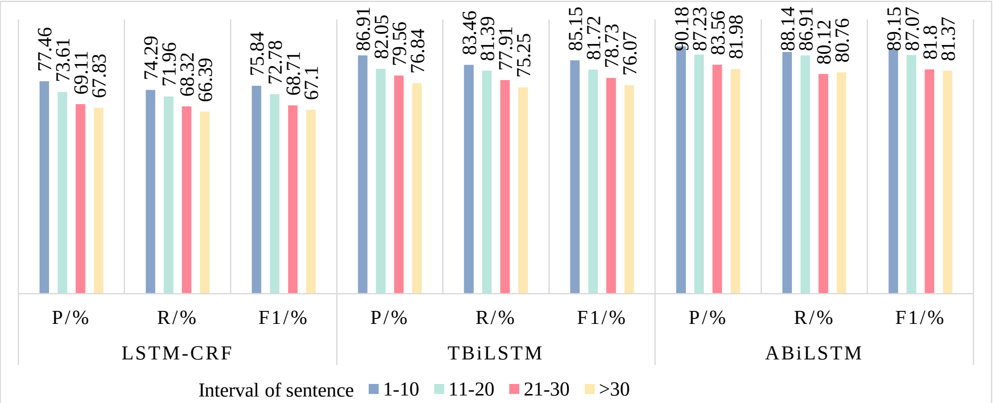

Figure 7. Comparison of CSRA performance for different sentence lengths

As can be seen from Figure 7, the F1 values of the three models on the test sets of the four sentence length intervals in Chinese decrease with the increase of sentence length, which shows that the longer the sentence length is, the more difficult it is for each model to perform semantic role annotation. On the short sentence test set, e.g., the sentence length interval of 1-10, the F1 values of LSTM-CRF, TBiLSTM, and BiLSTM are 75.84%, 85.15%, and 89.15%, respectively, and the F1 values of ABiLSTM are improved by 13.31% and 4% compared to LSTM-CRF and TBiLSTM, respectively. As for the long sentence test set, for example, the sentence length interval of >30, the F1 values of LSTM-CRF, TBiLSTM, and BiLSTM are 67.1%, 76.07%, and 81.37%, respectively, and the F1 values of ABiLSTM are improved by 14.27% and 5.3% in comparison with those of LSTM-CRF and TBiLSTM, respectively. This is because when a one-way LSTM cell establishes semantic associations for words that are far away from each other in a sentence, it needs to calculate the corresponding time step according to the distance of the words, which may result in the loss of semantic information. On the other hand, two-way LSTM combined with the attention mechanism can establish a direct semantic link for any word in the sentence with a single time step, which improves the model's ability to model the semantic information of long sentences, and thus improves the performance of semantic role annotation on long sentences.

6. **Conclusion.** Semantic role annotation is an intermediate step in many NLP tasks. Currently, deep learning methods are mainly used, but they need to calculate the corresponding time step according to the sentence length, which makes it difficult to establish semantic links between words that are far away from each other in long sentences, resulting in low performance on long sentences. For this reason, this paper proposes a CSRA method based on BiLSTM. Due to the complexity of Chinese sentences, the distribution of semantic role labels in the dataset is very uneven. In order to address this problem, we use association learning to select sentences similar to the target sentence from the annotated dataset as associative sentences, and use the associative sentences and their labels to help the target sentence complete the semantic role labelling in a better way, without the introduction of external resources. Then, Bi-LSTM encoder is used to encode the context of each word in the sentence, and the Self-Attention mechanism is introduced to model the semantic importance of each word in the sentence. The semantic role annotation results are then obtained after global normalization using CRF. The experiments imply that the designed approach can accurately annotate the semantic roles of Chinese sentences with the F1 values of 5.3%-14.27% higher than the comparison model in long sentence intervals.

REFERENCES

[1] H. Zhuang, C. Wang, C. Li, Y. Li, Q. Wang, and X. Zhou, "Chinese language processing based on stroke representation and multidimensional representation," IEEE Access, vol. 6, pp. 41928-41941, 2018.

[2] J. Yan, and H. Liu, "Semantic roles or syntactic functions: The effects of annotation scheme on the results of dependency measures," Studia Linguistica, vol. 76, no. 2, pp. 406-428, 2022.

[3] H. Yang, and C. Zong, "Learning generalized features for semantic role labeling," ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), vol. 15, no. 4, pp. 1-16, 2016.

[4] R. Cai, and M. Lapata, "Syntax-aware semantic role labeling without parsing," Transactions of the Association for Computational Linguistics, vol. 7, pp. 343-356, 2019.

[5] D. Gildea, and D. Jurafsky, "Automatic labeling of semantic roles," Computational Linguistics, vol. 28, no. 3, pp. 245-288, 2002.

[6] N. Xue, "Labeling Chinese predicates with semantic roles," Computational linguistics, vol. 34, no. 2, pp. 225-255, 2008.

[7] F. Wan, X. He, D. Zhang, G. Qi, A. Zhu, Z. Lei, N. Zenan, and W. Yicheng, "Chinese Shallow Semantic Parsing Based on Multi-method of Machine Learning," Journal of Web Engineering, vol. 19, no. 5-6, pp. 685-706, 2020.

[8] Y. Zhou, and N. Xue, "The Chinese Discourse TreeBank: A Chinese corpus annotated with discourse relations," Language Resources and Evaluation, vol. 49, pp. 397-431, 2015.

[9] Y.-N. Wang, X. Tian, and G. Zhong, "FFNet: Feature fusion network for few-shot semantic segmentation," Cognitive Computation, vol. 14, no. 2, pp. 875-886, 2022.

[10] Y.-C. Wang, F.-C. Wan, and N. Ma, "Multi-clue Chinese semantic role labeling based on conditional random fields," Journal of Yunnan University: Natural Sciences Edition, vol. 42, no. 3, pp. 474-480, 2020.

[11] B. Yu, and Z. Fan, "A comprehensive review of conditional random fields: variants, hybrids and applications," Artificial Intelligence Review, vol. 53, no. 6, pp. 4289-4333, 2020.

[12] Z. Huang, and Y. Chen, "An integration model of semantic annotation based on synergetic neural network," Intelligent Automation & Soft Computing, vol. 22, no. 3, pp. 525-532, 2016.

[13] Y. Shen, Y. Mai, X. Shen, W. Ding, and M. Guo, "Jointly part-of-speech tagging and semantic role labeling using auxiliary deep neural network model," Computers, Materials & Continua, vol. 65, no. 1, pp. 529-541, 2020.

[14] H. Song, S. Wang, Y. Liu, and Y. Wang, "Predicate-attention neural model for Chinese semantic role labeling," Computers and Electrical Engineering, vol. 99, 107741, 2022.

[15] J. C.-W. Lin, Y. Shao, Y. Djenouri, and U. Yun, "ASRNN: A recurrent neural network with an attention model for sequence labeling," Knowledge-Based Systems, vol. 212, 106548, 2021.

[16] F. A. Gers, and E. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," IEEE Transactions on Neural Networks, vol. 12, no. 6, pp. 1333-1340, 2001.

[17] Y. Jin, J. Xie, W. Guo, C. Luo, D. Wu, and R. Wang, "LSTM-CRF neural network with gated self attention for Chinese NER," IEEE Access, vol. 7, pp. 136694-136703, 2019.

[18] Y. Wang, Z. Lei, and W. Che, "Character-Level Syntax Infusion in Pre-Trained Models for Chinese Semantic Role Labeling," International Journal of Machine Learning and Cybernetics, vol. 12, pp. 3503-3515, 2021.

[19] X. Su, R. Li, X. Li, B. Chang, Z. Hu, X. Han, and Z. Yan, "A Span-based Target-aware Relation Model for Frame-semantic Parsing," ACM Transactions on Asian and Low- Resource Language Information Processing, vol. 22, no. 3, pp. 1-24, 2023.

[20] S. F. Ahmed, M. S. B. Alam, M. Hassan, M. R. Rozbu, T. Ishtiak, N. Rafa, M. Mofijur, A. Shawkat Ali, and A. H. Gandomi, "Deep learning modelling techniques: current progress, applications, advantages, and challenges," Artificial Intelligence Review, vol. 56, no. 11, pp. 13521-13617, 2023.

[21] J. Du, C.-M. Vong, and C. P. Chen, "Novel efficient RNN and LSTM-like architectures: Recurrent and gated broad learning systems and their applications for text classification," IEEE Transactions on Cybernetics, vol. 51, no. 3, pp. 1586-1597, 2020.

[22] L. Tan, and D. Taniar, "Adaptive estimated maximum-entropy distribution model," Information Sciences, vol. 177, no. 15, pp. 3110-3128, 2007.

[23] B. Mor, S. Garhwal, and A. Kumar, "A systematic review of hidden Markov models and their applications," Archives of Computational Methods in Engineering, vol. 28, pp. 1429-1448, 2021.

[24] L.-L. Liu, Y.-M. Cheng, and S.-H. Zhang, "Conditional random field reliability analysis of a cohesion-frictional slope," Computers and Geotechnics, vol. 82, pp. 173-186, 2017.

[25] G. P. Urcelay, and R. R. Miller, "The functions of contexts in associative learning," Behavioural Processes, vol. 104, pp. 2-12, 2014.

[26] S. P. Chatzis, and Y. Demiris, "The infinite-order conditional random field model for sequential data modeling," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 6, pp. 1523-1534, 2012.

[27] L. Zhang, T. Geisler, H. Ray, and Y. Xie, "Improving logistic regression on the imbalanced data by a novel penalized log-likelihood function," Journal of Applied Statistics, vol. 49, no. 13, pp. 3257-3277, 2022.

[28] Y. Tian, Y. Zhang, and H. Zhang, "Recent advances in stochastic gradient descent in deep learning," Mathematics, vol. 11, no. 3, 682, 2023.

[29] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection," IEEE Transactions on Wireless Communications, vol. 19, no. 5, pp. 3319-3331, 2020.

[30] X. Bai, and N. Xue, "Generalizing the semantic roles in the Chinese Proposition Bank," Language Resources and Evaluation, vol. 50, pp. 643-666, 2016.