Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

A Study of Using Principal Component Analysis to Enhance Semantic Role Annotation in Chinese Autoanalysis

Geng Wang[1], Gangqiang Li[2,] ✉

[1] *Department of Chinese, School of Culture and Media, Huanghuai University, Zhumadian, Henan, 463000, China*

[2] *Department of Information and Communication Engineering, School of Computer and Artificial Intelligenc, Huanghuai University, Zhumadian, Henan, 463000, China*

ABSTRACT

The article explores the ways to improve the effect of semantic role labeling in the process of automatic Chinese language analysis. The traditional calculation methods for information content, distance and attributes are improved, and the dynamic weight calculation method for semantics is proposed and comprehensively weighted based on principal component analysis. A Chinese semantic role labeling model based on Highway Bi-LSTM is designed, and Self-Attention is added to improve the information representation of semantic role labeling. The correlation coefficient of this paper's method with the manual judgment value is above 0.9, and the average Kappa coefficient is 82.43%, which is very close to the effect of manual annotation. The F1 values of this paper's model are improved by 18.56% and 8.95% respectively after adopting the attention mechanism, indicating that the design of this paper's method is reasonable. It can be seen that the method of this paper helps to improve the effect of semantic role labeling.

Keywords: principal component analysis, Highway Bi-LSTM, semantic role labeling, automatic Chinese language analysis

1. Introduction

Semantic analysis is an important task in the field of natural language processing, which is devoted to obtaining the semantic information embedded in a given sentence and representing it in a way that computers can understand [1-2]. Currently, the research work on semantic analysis mainly includes deep semantic analysis and shallow semantic analysis. The main work of deep semantic analysis includes pervasive concept cognitive labeling, semantic dependency analysis, and abstract semantic representation. However, deep semantic analysis has the problems of wide range of semantic levels, ambiguity and vagueness, difficulty in portraying all the semantic information with a good

✉ Corresponding author.
E-mail address: Wanggeng_1125@163.com (G. Li).
Received 25 March 2024; Revised 05 June 2024; Accepted 30 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-240

formalization method, and even limited by the current technical level, it is not easy to form the results with strong practicality in the short term [3-6]. In terms of shallow semantic analysis, the current main realization is semantic role annotation, which has many advantages such as multi-language generalization and natural presentation [7-10].

Semantic role annotation, as the most common method to realize shallow semantic analysis, aims to annotate all the semantic roles related to predicates in a sentence, such as the giver, the receiver, the time and the place, etc. [11-12]. Semantic role annotation is characterized by concise annotation, clear structure, easy presentation, and feasible technology, which has a wide range of applications in the fields of automatic question and answer, machine translation, reading comprehension, and information extraction, and is of great significance for future research on deep semantic analysis as well as chapter understanding [13-17]. Compared with English semantic role annotation, Chinese semantic role annotation started relatively late, the annotation corpus is not perfect, and Chinese has unique linguistic characteristics, such as varied inflectional changes, complex grammatical functions, etc., so there is still a lot of room for improvement and optimization in the existing Chinese semantic role annotation model [18-22]. Therefore, it is very necessary to study the Chinese semantic role annotation technology in depth, so as to improve the annotation effect of the Chinese semantic role annotation model.

In this paper, the information content, distance and attribute calculation methods in Chinese automatic analysis are improved respectively. The information content of the ontology concept itself is utilized for solving, the semantic distance is calculated based on the geometric distance between two concept words, and the semantic similarity calculation of the concepts is completed by the number of common attributes. The dynamic weights of semantics are calculated and comprehensively weighted using principal component analysis. On this basis, a Chinese semantic role labeling model based on Highway Bi-LSTM is proposed. Bi-LSTM is taken as the core component of the model, and Highway-based Bi-LSTM technique is adopted to solve the gradient vanishing and explosion problems. Finally the embedded Self-Attention mechanism enriches the information representation of High way Bi-LSTM to enhance the semantic role labeling effect. Semantic similarity calculation and semantic role labeling accuracy experiments are carried out to compare and verify the superiority of this paper's method over existing semantic role labeling methods.

2. Dynamic weight calculation method based on principal component analysis

In the automatic analysis of Chinese language, three main influencing factors are targeted in the calculation of ontological semantic similarity: conceptual information content, conceptual attributes and conceptual distance. In this paper, we analyze the shortcomings in the traditional calculation methods and improve the calculation methods based on information content, distance and attributes respectively.

2.1. Ontology-based multifactor semantic similarity computation

2.1.1. Calculation methods based on information content. The information content-based computation method mainly calculates the information content by utilizing the information shared between two concepts in a particular ontology to complete the determination of the semantic similarity between concepts. For calculating the semantic similarity between any two concepts in an ontology can be shown in equation (1):

$$Sim(a,b)_{ic} = \frac{2IC(Lcan(a,b))}{IC(a)+IC(b)} \tag{1}$$

Where $Lcan(a,b)$ denotes the nearest neighbor common ancestor node of concepts a and b in the ontology tree, and $IC(a)$ and $IC(b)$ denote the information content contained in concepts a

and b, generally referred to as the amount of information, which provides a way to calculate the semantic similarity by quantitatively representing the information, and the formula for the amount of information is shown in (2):

$$IC(c) = -\log\frac{N(c)}{N} \tag{2}$$

where $N(c)$ is the number of occurrences of concept c in the training set and N is the total number of training sets. Its calculation results depend on the number of training samples in the corpus, so this method of calculating the information content of concept nodes using the corpus has shortcomings [23]. By analyzing the ontology hierarchy, this paper argues that the main factors affecting the information content of concepts and the semantic similarity between concepts are (1) the depth at which the concepts being computed are located in the ontology hierarchy. (2) The density of the region where the computed concepts are located in the ontology hierarchy. (3) The length of the paths separating the computed concepts in the ontology hierarchy.

Therefore, this paper proposes a method of solving the problem by utilizing the amount of information of the ontology concept itself, as shown in equation (3):

$$IC(\mathrm{c}) = \left(\frac{mNode(c)}{mNode(T)} + \frac{\log(D(c))}{\log(mD(T))}\right) * \left(1 - \frac{\log(h(c)+1)}{\log(mNode(T))}\right) \tag{3}$$

Where, $mNode(c)$ is the total number of concept nodes owned by the subtree with the root node as the direct parent node where the node of concept c is located, $mNode(T)$ is the number of all concept nodes in the ontology tree T where the concept c is located, $h(c)$ is the number of all child nodes of concept c, $D(c)$ is the depth of concept c, and $mD(T)$ is the maximum depth of the ontology tree T. The information content of the concepts is found and then the similarity of the two concepts is calculated using equation (1).

2.1.2. Distance-based calculation methods. The distance-based computational approach utilizes the geometric distance between two conceptual words in the ontology hierarchy to compute the semantic distance between them, and the computational model is shown in Equation (4):

$$Sim(a,b) = \frac{1}{Dis(a,b)} \tag{4}$$

Where $Dis(a,b)$ denotes the shortest distance between concepts. W-P and L-C are two classical algorithms in semantic distance-based ontology semantic similarity computation, but both of them ignore the effect of the type of relationship of the edges on the computation Different types of relationships have different effects on the path. Currently, there are more methods that consider the types of edges in the study of distance-based similarity computation, but most of them do not target the types of edges to be the main factor in the computation [24].

In this paper, the type of relationship of the edges is added as weights to the calculation process, and the weights of the corresponding edges are defined for the three main conceptual relationships in the ontology tree as shown in Equation (5):

$$Wedge(a,b) = \begin{cases} 0.9 & is\ A \\ 0.5 & partof \\ 0.1 & other \end{cases} \tag{5}$$

Combined with the W-P algorithm, an improved computational model such as (6) is proposed:

$$Sim(a,b)_{dis} = \frac{2 * sPath(c,r) + sPath(a,b)}{sPath(a,b) + sPath(b,c) + 2 * sPath(c,r)} \tag{6}$$

The inter-conceptual weighted shortest path is calculated as in (7):

$$sPath(a_1,a_n)=\sum_{i=I}^{n} wedge_i^{*}\, path(a_i,a_{i+I}) \tag{7}$$

where $wedge$ is the weight of the edges between neighboring concepts, $path$ is the direct distance between neighboring concepts, $sPath(a,b)$ denotes the weighted shortest path from concept α to concept b, $sPath(a,c)$ and $sPath(b,c)$ denote the shortest paths from concepts α and b, respectively, to the nearest common node c, and $sPath(c,r)$ denotes the shortest path from concept c to the root node r.

2.1.3. Attribute-based. calculation methods Ontological concepts are characterized by concept attributes, and attribute-based computational methods utilize counting the number of public attributes that a concept possesses to complete the semantic similarity computation of the concepts. The similarity of concepts is proportional to the number of common attributes possessed by the concepts. The classical attribute-based semantic similarity calculation method is shown in equation (8):

$$\begin{aligned} Sim(a,b)=&\alpha\times \Pr operties(a\cap b)\\ &-\beta\times \Pr operties(a-b)\\ &-\gamma\times \Pr operties(b-a) \end{aligned} \tag{8}$$

Where $Properties(a\cap b)$ denotes the set of common attributes owned by concepts α and b, $Properties(a-b)$ denotes the set of attributes owned by concept α but not owned by concept b, and $Properties(b-a)$ denotes the set of attributes owned by concept b but not owned by α.

In this paper, an improved calculation method is proposed by combining the attribute structure information as shown in equation (9):

$$Sim(a,b)_{pro}=\frac{\Pr operties(a\cap b)}{\Pr operties(a\cap b)+\alpha^{*}\Pr operties(a-b)+\beta^{*}\Pr operties(b-a)} \tag{9}$$

The structure of an attribute affects the attribute through the conceptual depth, which is formulated as in (10):

$$a=\begin{cases} \dfrac{d(a)}{d(a)+d(b)} & d(a)\le d(b)\\ I-\dfrac{d(a)}{d(a)+d(b)} & d(a)>d(b) \end{cases} \qquad \alpha+\beta=1 \tag{10}$$

where $Properties(a\cap b)$ denotes the set of common attributes possessed by concepts a and b, $Properties(a-b)$ denotes the set of attributes possessed by concept a and not possessed by concept b, and $Properties(b-a)$ denotes the set of attributes possessed by concept $Q_{key}=\{q_{1,}q_{2,}\cdots,q_i,\cdot,q_n\}(i=1,2,...,n)$ and not possessed by concept a. $d(a)$ and $d(b)$ are the depths of concepts a and b in the ontology hierarchy, respectively.

2.2. Semantic Dynamic Weights Calculation Method Based on Principal Component Analysis

The weights of the principal components in this paper are assigned according to their contribution rate to ensure the objectivity, rationality and accuracy of the results, and the calculation process is as follows [25]:

1) Elimination of the influence of different variables of the scale, first standardized variables. Let there be a total of η sample of test data and P indicators, set variable X_i,X_j,X_p, and let $X_{ij}(i=1,2\cdots,n;j=1,2,\cdots,p)$ be the value of the j th indicator of the i th sample. Make the transformation:

$$Y_j = \frac{X_j - E(X_j)}{\sqrt{Var(X_j)}} \quad (i, j = 1, 2, \cdots, p) \tag{11}$$

Obtain the standardized data matrix $y_{ij} = \frac{x_{ij} - \overline{x}_j}{s_j}$, where $\overline{x}_j = \frac{1}{n}\sum_{i=1}^{i} x_{ij}, s_j^2 = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2$.

2) Calculate the matrix of correlation coefficients of the p original indicators using the standardized data matrix $Y = (y_{ij})n * p$:

$$R = (r_{ij})_{p\times p} = \begin{bmatrix} \eta_{11} & \eta_{12} & \cdots & \eta_{1P} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ r_{p1} & r_{p2} & \cdots & r_{pp} \end{bmatrix} \tag{12}$$

Among them:

$$r_{ij} = \frac{\sum_{k=1}^{n}(x_{ki} - \overline{x}_i)(x_{kj} - \overline{x}_j)}{\sqrt{\sum_{k=1}^{n}(x_{ki} - \overline{x}_i)^2 \sum_{k=1}^{n}(x_{kj} - \overline{x}_j)^2}} (i, j = 1, 2, \cdots, p) \tag{13}$$

3) Solve the eigenvalues of the correlation coefficient matrix R and sort $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$, then find the regularized eigenvector $e_i = (e_{i1}, e_{i2}, \ldots, e_{ip})$ corresponding to the eigenvalues of R, then the ith principal component is expressed as a combination $Z_i = \sum_{k=1}^{p} e_{ik} \cdot X_k$ of each index X_k.

4) Calculate the cumulative contribution ratio and determine the number of principal components. The contribution rate of principal component Z_i is:

$$w_i = \frac{\lambda_i}{\sum_{k=1}^{p}\lambda_k} (i = 1, 2, \cdots, p) \tag{14}$$

The cumulative contribution is:

$$\frac{\sum_{k=1}^{i}\lambda_k}{\sum_{k=1}^{p}\lambda_k} (i, j = 1, 2, \cdots, p) \tag{15}$$

Count the 1st, 2nd, ..., $m(m \leq p)$ nd principal components corresponding to the eigenvalue $\lambda_1, \lambda_2, \cdots, \lambda_m$ with a contribution rate of 85%~95%.

5) Calculate the principal component loadings to determine the composite score. The principal component loading is that the larger the correlation coefficient between the principal component and each index, the better the representation of the principal component for the index variable, and the calculation formula is:

$$l_{ij} = p(z_i, x_j) = \sqrt{\lambda_i} e_{ij} (i, j = 1, 2, \cdots, p) \tag{16}$$

6) Determine the score of each principal component and determine the composite scoring function. After obtaining the loadings of each principal component, the score of each principal component can be calculated:

$$\begin{cases} z_1 = l_{11}\, x_1 + l_{12}\, x_2 + \cdots + l_{1p}\, x_p \\ z_2 = l_{21}\, x_1 + l_{22}\, x_2 + \cdots + l_{2p}\, x_p \\ \cdots \\ z_m = l_{m1}\, x_1 + l_{m2}\, x_2 + \cdots + l_{mp}\, x_p \end{cases} \tag{17}$$

$$Z = (z_{ij})_{n\times m} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nm} \end{bmatrix} \tag{18}$$

where Z_{ij} denotes the jrd principal component score of the ind sample, then the composite score of the ith sample:

$$f_i = \sum_{k=1}^{m} w_k \cdot z_{ik} (i = 1, 2, \cdots, n) \tag{19}$$

In this paper, the contribution of each factor is calculated as weights using principal component analysis. The original principal component analysis method is to determine the principal components by the cumulative contribution rate is greater than a set threshold. The three factors proposed in this paper: the amount of information, distance, and attributes are to be used as principal components, which can be ignored to improve the efficiency of the algorithm. The main ideas of the dynamic weight calculation method in this paper are as follows:

(1) The three similarities calculated from the three factors are used as three dimensions, and the similarity matrix is obtained as the input sample matrix through the calculation of multiple samples.

(2) Perform a matrix normalization transformation of the sample matrix to the standard matrix Z and find the correlation coefficient matrix R.

(3) Find the 3 characteristic roots of the characteristic equation of the sample correlation matrix R to determine the principal components $\lambda_1, \lambda_2, \lambda_3$.

(4) Solve the system of equations $R*b = \lambda_i * b_{(i=1,2,3)}$ unit eigenvectors b_i^o.

(5) Convert the standardized indicator variables into principal components $U_i = Z_i^T * b_i^o (i = 1, 2, 3)$

(6) Weight and linearly sum the 3 principal components to obtain the final semantic similarity value, and the weights are the contribution of each principal component.

2.3. Comprehensive weighting calculations

The calculation method of comprehensive weighting involves the setting of weights, and in the past, experts were manually assigned in the process of determining the weights, which made the results subjective uncertainty, and the determination of the weights of ontologies in different domains also needs to be determined by experts in different domains, which makes the calculation of the weights not self-adaptive. In this paper, after considering the three factors of concept information, distance and attributes to calculate the similarity degree respectively, the three similarities are dynamically weighted linearly summed with the improved principal component analysis. If the number of concept pairs in the set of compared concept pairs is m, and $X_i = (Sim_{ic(i)}, Sim_{diz(i)}, Sim_{pro(i)})$ is set as a vector in the set of principal component input samples, in which each dimensional variable represents the result of semantic similarity computation of each part in the comprehensive similarity computation module, then the concept semantic similarity matrix is denoted as $X_{\sin} = (x_{i1}, x_{i2}, x_{i3})^T, i = 1, 2, \ldots\ldots, m$. The constructed semantic similarity matrix $X_{\sin}$ is subjected to principal component analysis, and the extracted principal component is $Y = (y_{sim1}, y_{sim2}, y_{sim3})$, and the contribution rate of each principal component is (r_1, r_2, r_3). Then the final concept semantic similarity calculation formula is shown in equation (20):

$$Sim_{total} = r_1 * y_{sim1} + r_2 * y_{sim2} + r_3 * y_{sim3} \tag{20}$$

3. Highway Bi-LSTM-based semantic role labeling model for Chinese language

Applying the idea of principal component analysis to the work of Chinese semantic role labeling, a Chinese semantic role labeling model based on Highway Bi-LSTM is proposed.

3.1. Core layer structure

Figure 1 shows the framework of the Chinese semantic role model, and the core layer structure is shown in the wireframe part of Figure 1, which can be seen to be composed of two parts of molecular layers. One part adopts the Highway Bi-LSTM technique and the other part adopts the Self-Attention technique. The principles and effects of these two techniques are explained in detail below, respectively.

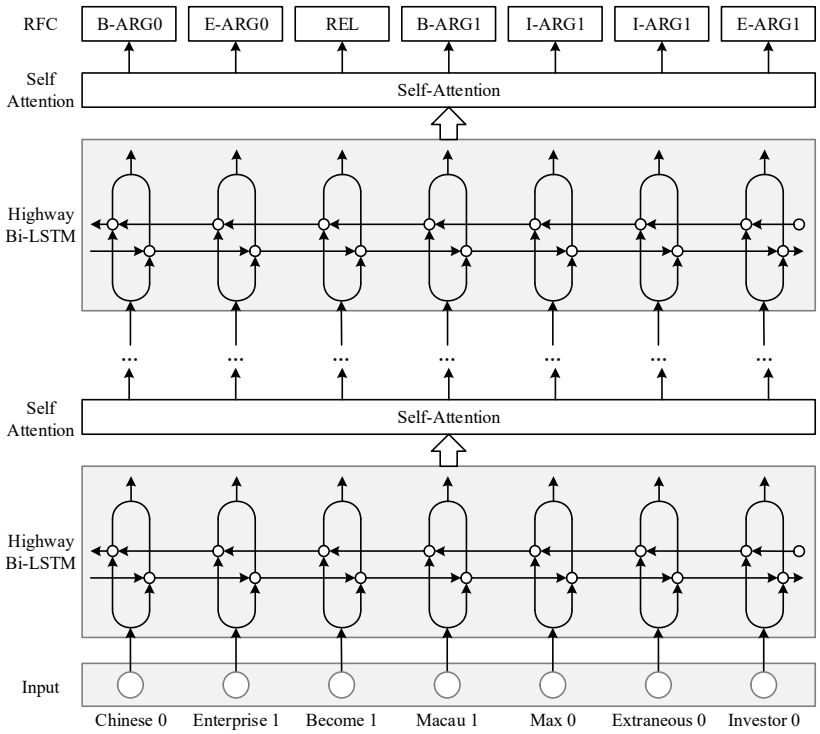

Fig. 1. Framework of Chinese semantic role model

3.2. Deep High way Bi-LSTM

We use a bidirectional LSTM as the core component of the model. In contrast to GRU, LSTM utilizes three gates for information control. These are the forgetting gate, the input gate, and the output gate. The forgetting gate is analogous to the reset gate in GRU and is able to selectively discard information that the model considers redundant. The input gate can filter the model input for new features, which is not available in the GRU model. Input gates and forgetting gates correspond to the weights of the current feature information and the previous moment's state information, respectively. The output gate is used to determine the final output of the model. The corresponding formulas are shown below:

$$i_{l,t} = \sigma\left(W_i^l\left[h_{l,t-1}, x_{l,t}\right] + b_i^l\right) \tag{21}$$

$$o_{l,t} = \sigma\left(W_o^l\left[h_{l,t-1}, x_{l,t}\right] + b_o^l\right) \tag{22}$$

$$f_{l,t} = \sigma\left(W_f^l\left[h_{l,t-1}, x_{l,t}\right] + b_f^l + 1\right) \tag{23}$$

$$\tilde{c}_{l,t} = \tanh\left(W_c^l\left[h_{l,t+1}, x_{l,t}\right] + b_c^l\right) \tag{24}$$

$$c_{l,t} = i_{l,t} \circ \tilde{c}_{l,t} + f_{l,t} \circ c_{l,t-1} \tag{25}$$

$$h_{l,t} = o_{l,t} \circ \tanh(c_{l,t}) \tag{26}$$

$i_{l,t}$ is the input gate at moment t of layer l, $o_{l,t}$ is the output gate at moment t of layer l, and $f_{l,t}$ is the forgetting gate at moment t of layer l. It can be seen that they all consist of the input of the current moment and the output of the previous moment, which are linearly mapped with the corresponding weights, and then the final result is mapped to [0, 1] using a sigmoid function. Equation (24) represents the feature information representation of the input at moment t, and equation (25) represents the corresponding feature information at moment t, which can be seen as a combination of input gate and forget gate by filtering the current input information and the information of the previous moment, and this state will be passed into the calculation of the new state at the next moment. Equation (26) represents the final output of the current moment.

Because the unidirectional network can only learn the dependency information in the sentence from one direction, but the components in the sentence are not only related to the previous content, but also have a strong connection with the subsequent content. Therefore, in this paper, bidirectional LSTM is applied to learn features from two directions. The specific implementation is shown in the following equation:

$$\overrightarrow{h_t} = LSTM(x_t, h_{t-1}) \tag{27}$$

$$\overleftarrow{h_t} = LSTM(x_t, h_{t+1}) \tag{28}$$

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{29}$$

x_t denotes the t nd word in the sentence, and $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are the cryptohan states extracted in both directions before and after x_t, with the same dimensions. Since $\oplus$ represents a connective here, the length of h_t is twice that of $\overrightarrow{h_t}$.

Semantic character annotation is a more complex task, and it is difficult to fully mine the feature information present in the training samples with only a single layer neural network. In this paper, the number of network layers is increased, i.e., the depth of the model is increased from a spatial point of view to improve the model's annotation capability. However, deepening the number of network layers will directly lead to the problem of gradient vanishing or explosion. Based on this, this paper uses a bi-directional LSTM technique based on Highway connection. This method is similar to the implementation principle of residual networks, which combines a certain percentage of the input portion on top of the previous results in order to prevent the loss of information [26]. Similar to the gating system in LSTM or GRU, it utilizes a component known as a “transition gate” to control how much of the original information is retained. In this case, the output of the LSTM has a new twist on the original, as shown in the following equation:

$$newh_{l,t} = T \times h_{l,t} + (1-T) \times x_{l,t} \tag{30}$$

Where T denotes the “conversion gate”, $h_{l,t}$ is the output of the Bi-LSTM in layer l, moment t, and $newh_{l,t}$ is the new output after using the conversion gate.

3.3. Self-Attention mechanism

Self-Attention is an attentional mechanism for dealing with inter-elements within a sequence, which can directly portray the global dependencies of an entire sentence. The biggest advantage of this

mechanism is that it can acquire the dependency between any two words in a sentence without considering the distance, and at the same time learn the internal structural information in the sentence. It maps out the feature information in a sentence in an easier and more flexible way. The mechanism is combined with deep High way Bi-LSTM as a complement to High way Bi-LSTM to get richer information representation.

Assuming that the Self-Attention mechanism is to be applied to a particular sentence that has been divided into words, one can start by assuming that each word in the sentence consists of a series of <Key, Value> pairs. To process one of these words, the relevance of all the words in the sentence to it needs to be computed, and the result obtained is called the Attention Value. The first step in the implementation of the mechanism is to generate three vectors for each word corresponding to $Query_t(Q_t), Key_t(K_t), Value_t(V_t)$. The subscript t indicates that this is the result of three computations for the trd word. This part is obtained by multiplying three weight vectors (W_{qt}, W_{kt}, W_{vt}) using a vector H_t respectively, these weight vectors are the parameters that the model needs to be trained on, here H_t is the result of the execution of the t th time step in High way Bi-LSTM, the implementation is shown in the following equation:

$$H \times W_{q,t} = Q_t \tag{31}$$

$$H \times W_{k,t} = K_t \tag{32}$$

$$H \times W_{v,t} = V_t \tag{33}$$

The next step is to use the current word to be processed as the target vector Query of the query, to go and perform similarity computation with the Key vectors of all the words in the sentence (including the word itself). Assuming that the attention value between the ist word and each other word is to be computed, Q_i represents the target vector corresponding to this word, and K_j represents the Key vector corresponding to the jth word, the result of the dot product of Q_i and each of K_j is to be computed, and the value of j is in the range of $j \in \{1,2,3,..M\}$, and M is the length of the sentence. Scale scaling, and normalization operations are performed on the computed results to obtain the corresponding attention values. Finally, these attention values are multiplied with the corresponding Value vectors to obtain the weighting vector of the current word. In practice, in order to speed up the computation, matrix computation is often used, i.e., each input vector is combined into a matrix, and the corresponding weights are also combined into a matrix. The above process can be expressed by the following equation:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{34}$$

d represents the dimension of K or Q, and dividing the result of multiplying Q and K by the root sign d is the scale scaling operation performed, which aims at obtaining a more stable gradient and preventing the occurrence of gradient vanishing or explosion.Softmax operation is performed in order to get the corresponding weight value of each word in the sentence.

In order to enhance the model's ability to focus on different positions and enrich the representation of the subspace, this paper also adopts the multi-head attention mechanism, which in essence uses multiple sets of Q, K, V vectors to perform multiple computations. Using 8 sets of vectors for training, the computed results of multiple Attention are merged into a single matrix, and then a fixed dimension result is obtained by multiplying with a weight matrix. The corresponding multiple Attention mechanism can be represented by the following equation:

$$head_i = Attention(QW_i^q, KW_i^k, VW_i^v) \tag{35}$$

$$Head' = (head_1 \oplus \cdots \oplus head_n) \tag{36}$$

$head_i$ denotes the result of the ind Attention, and $\oplus$ denotes the connection operation.

3.4. Decoding layer

In this paper, we call the above neural network module as the Encoder part of the model, whose role is mainly to perform feature extraction on the input sequence data, and the generated data structure is still a vector sequence. However, we want the model to output one discrete labeled result, therefore, this paper sets the conditional random field as the final decoding layer algorithm.

Conditional Random Field is a classical machine learning algorithm. In decoding, it considers sentence-level labeling information and applies a Viterbi algorithm to obtain the optimal labeling sequence. For example, suppose a sentence X, with $\{x_1,\cdots x_n\}$ representing the symbolic input, x_i representing the vector representation of the ith word in the sentence, and defining a role sequence $y=\{y_1\cdots y_n\}$, where y_i represents the role of the ith word in the sentence, the CRF will start from the whole sentence, compute a score for each possible label sequence, and then normalize the probability of generating the individual label sequences by a Softmax function. The realization principle is as follows:

$$s(X,y)=\sum_{i=0}^{n}A_{y_i,y_{i+1}}+\sum_{i=1}^{n}P_{i,y_i} \tag{37}$$

$$P(y\mid X)=\frac{e^{s(X,y)}}{\sum_{\bar{y}\in Y_x}e^{s(X,\bar{y})}} \tag{38}$$

where $s(X,y)$ corresponds to the score when the output sequence is y, $A\in R^k\times k$ represents here the role transfer matrix, and k is the total number of types of roles. $A_{i,j}$ denotes the loss of transfer from role i to role j. $P\in R^n\times k$ is a role generation probability, and $P_{i,j}$ denotes the probability that the ith word in the sentence is defined as the jth role. In Eq. (38), Y_x denotes here the sequence of all possible roles corresponding to the sentence X of the input. $P(y\mid X)$ denotes the conditional probability that the output is y when the input is X and y the actual sequence of roles corresponding to X. The goal of the algorithm is to maximize $P(y\mid X)$. Therefore, the corresponding loss function is shown in Eq. (39). For the training of the model, the backpropagation algorithm can be used to maximize the log function of the character generation probability:

$$\log(P(y\mid X))=s(X,y)-\log(\sum_{\bar{y}\in Y_x}e^{s(X,\bar{y})}) \tag{39}$$

4. Example analysis

4.1. Semantic similarity calculation

In this paper, CoNLL2005 is used as the ontology semantic library to explore the problem of calculating the similarity between concepts. The JWNL (Java WordNet Library) software package is used to complete the parsing of WordNet, the multi-angle similarity calculation is completed using the Java language, and the process of principal component analysis and the analysis of the results are completed through the Matlab-oriented interface.

Firstly, a coordinate system is established using word pairs and corresponding similarity values as coordinates, and the distribution of similarity values calculated by this paper's algorithm and the classical algorithm is compared in the coordinate system. Figure 2(a)~(f) shows the distribution of 80 Chinese word pairs under Cosine, Jaccard, Edit, Word2Vec, BERT and this paper's algorithm, respectively.

From the result graphs, all the algorithms based on semantic distance have a high degree of discretization, and the continuity of the distributions obtained by this paper's algorithm is better,

which indicates that the similarity computed values based on this paper's algorithm have a good correlation with the manual scoring of similarity.

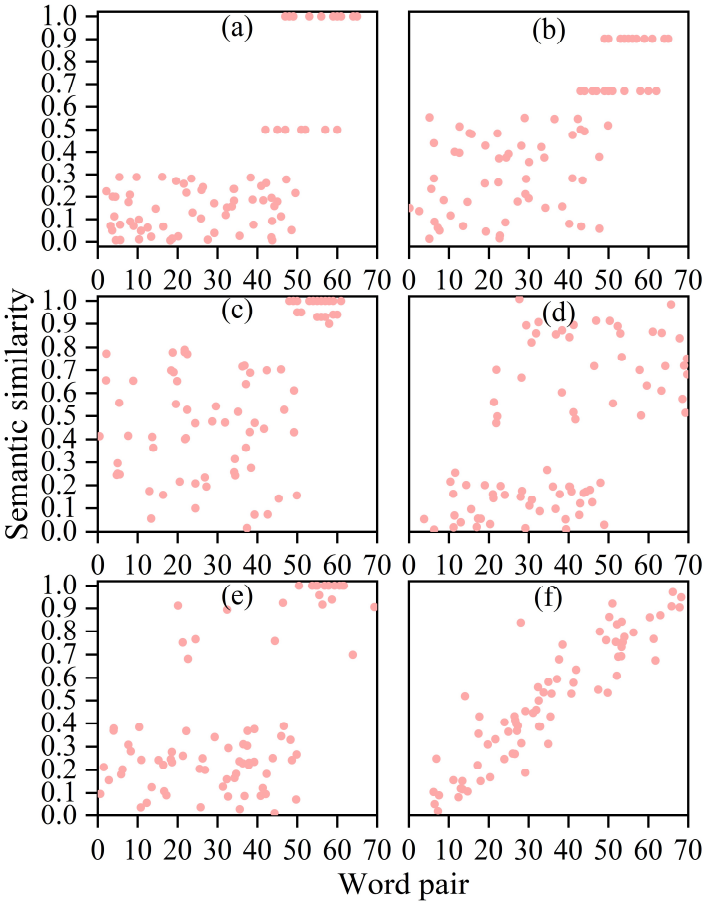

Fig. 2. The distribution of 80 Chinese word pairs in different algorithms

The correlation between various classical algorithms and this paper's algorithm with manual scoring is then utilized for comparison. The correlation coefficients between the similarity values under various algorithms and the manually scored values for similarity in the RG, P&S1 and P&S2 datasets are calculated respectively and the results are shown in Table 1.

From the results, it can be seen that the correlation coefficients between the similarity values calculated by the algorithm of this paper and the manually scored values of similarity in each dataset are maximized, so the accuracy of the similarity obtained by using the algorithm of this paper is high.

Table 1. The correlation coefficient of different algorithms and artificial judgment

Algorithm	RG	P&S1	P&S2
Cosine	0.77504	0.82748	0.83844
Jaccard	0.85265	0.87563	0.86978
Edit	0.82173	0.81515	0.81452
Word2Vec	0.80559	0.82344	0.83338
BERT	0.72777	0.76881	0.78643
Ours	0.90247	0.92025	0.9051

4.2. Semantic Role Annotation Accuracy

4.2.1. Analysis of model parameters. Firstly, all the data are preprocessed, i.e., a dependency tree is constructed for all the sentence sequences by an additional syntactic analyzer, and then the training data are inputted into the model for iterative training, and the relevant parameters of the model are continuously adjusted during training, so that the optimal model is obtained when the value of the training loss of the model reaches the minimum. The prediction stage inputs test data into the trained model and analyzes the final prediction results in comparison with other existing models.

When the model annotates the semantic roles of all the words in a sentence, only the best hyperparameters are set to make the annotation effect optimal. The selection of model parameters is determined by the results of a large number of experiments based on the control variable approach. In the training process, data batches are input into the model for training and the core parameters are continuously adjusted until the model loss value reaches the minimum, the optimal parameter combination of the model can be obtained.

In this paper, the other parameter values of the model are fixed first, and then the semantic role labeling dataset is input into the model for training, and the number of neurons in the hidden Han layer of the High way Bi-LSTM is adjusted continuously, and the loss value change curve obtained is shown in Fig. 3. Where the horizontal coordinate represents the number of training rounds of the model, and the vertical coordinate is the loss function value obtained in each round of training. In this paper, three commonly used settings for the number of neurons are selected: 100, 150, 200, from the loss function change curve in the figure, it can be seen that the number of cryptographic neurons needs to be set according to the distribution of the data, and it is not that the more the number of neurons corresponds to the better the effect, so the number of neurons in the High way Bi-LSTM cryptographic layer in this paper's model is set to 150 as the most appropriate.

Fig. 3. The effect of the number of neurons on the value of High way Bi-LSTM

In order to determine the number of neurons in the hidden Han layer of the bidirectional graph convolutional network, this paper first fixes the values of the other parameters in the model, and then constantly changes the set values of the hidden Han layer neurons, and the final change of the loss value of the bidirectional graph convolutional network based on the attention mechanism is shown in Figure 4. Where the horizontal coordinate represents the number of training rounds of the model, and the vertical coordinate is the loss function value obtained from each round of training. From the loss value change curve in the figure, it can be seen that, but when the number of neurons is set to 300, the model loss value has the smallest value in the steady state. So in this paper, 300 is chosen as the number of neurons in the hidden Han layer of the bidirectional graph convolutional network.

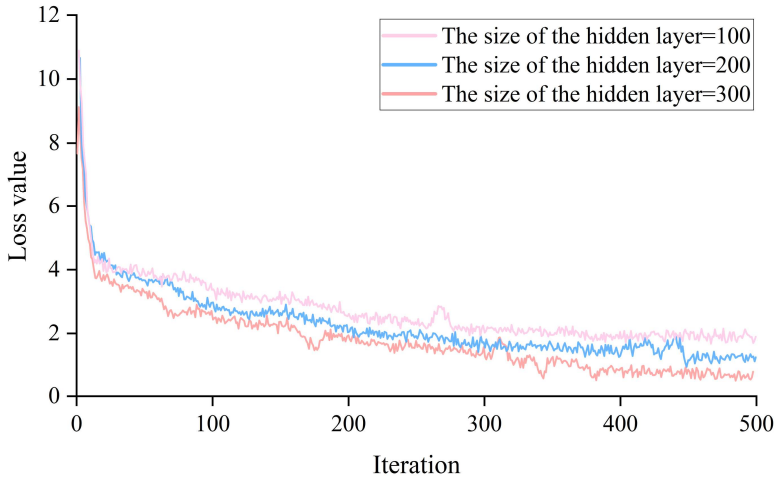

Fig. 4. The effect of the number of neurons on the value of High way Bi-GCN

For the other key parameters in the model, this paper adopts the same approach to determine the parameter values one by one, and the final settings of the present hyperparameters are shown in Table 2.

Table 2. The super parameter table

Parameter name	Parameter
Training time	50
Word embedded dimension	400
Lexical embedded dimensions	20
The number of neurons in the Bi-LSTM hidden layer	150
The number of neurons in the Bi-GCN hidden layer	300
Attention neuron number	60
Learning rate	0.0001
Batch size	128

4.2.2. Comparison of different models. The experimental models all use conditional random fields to decode the output to find the optimal labeling sequence for the model. The effect of the attentional mechanism on the experimental results is shown in Table 3.

With the same parameter settings, the F1 value of Bi-LSTM is improved by 3.85% compared to LSTM, which confirms the superiority of bi-directional long and short-term memory networks. After adding the attention mechanism to Bi-LSTM, the F1 values of the model on the two test sets were improved by 18.56% and 8.95%, respectively. This result suggests that the attention mechanism can select the dependency between predicates and arguments, further emphasizing the importance of predicates in semantic role labeling.

Table 3. The effect of the attention mechanism on the experimental results

Model	Verification set			Test set(WSJ)			Test set(Brown)		
	P	R	F1	P	R	F1	P	R	F1
LSTM	0.748	0.761	0.754	0.747	0.751	0.749	0.673	0.718	0.695
Bi-LSTM	0.791	0.776	0.783	0.762	0.737	0.749	0.722	0.730	0.726
Bi-LSTM + self-attention	0.877	0.864	0.870	0.902	0.875	0.888	0.800	0.782	0.791

In order to verify the effectiveness of the proposed model, this paper selects five existing excellent models to compare with it, and the experimental results are shown in Table 4, using the F1 value as the evaluation index of inter-model comparison.

By comparing and analyzing with the five models, it can be obtained:

1)The bidirectional long and short-term memory network effectively realizes the encoding of textual contextual information, and inputs the encoded word representation and predicate representation into the upper layer of the model after splicing, which facilitates the bidirectional graphic convolutional network to effectively utilize the lexical information in the secondary encoding process.

2)The attention mechanism emphasizes the importance of predicates in semantic role labeling through the selection of semantic dependencies between predicates and arguments.

3)The model decoding uses conditional random fields, which effectively ensures semantic constraints between semantic role labels.

The experimental results show that the model proposed in this paper has better performance and still achieves better results compared with the existing optimal models.

Table 4. Experimental results of different models

Model	Verification set			Test set(WSJ)			Test set(Brown)		
	P	R	F1	P	R	F1	P	R	F1
He et al.(2017)PoE	0.826	0.825	0.825	0.853	0.831	0.842	0.747	0.702	0.724
He et al.(2018)	-	-	0.813	0.811	0.843	0.827	0.703	0.707	0.705
He et al.(2018)+ELMo	-	-	0.860	0.853	0.863	0.858	0.746	0.794	0.770
Angel Daza et al.(2018)	-	-	0.776	-	-	0.808	-	-	0.682
Bi-GCN	-	-	-	-	-	0.878	-	-	0.790
Ours	0.879	0.850	0.865	0.899	0.882	0.891	0.799	0.789	0.794

4.3. Semantic Role Annotation Practices

For the actual annotation, this paper firstly selects 103871 Chinese sentences from primary and secondary Chinese textbooks and Chinese online texts as the main sources for semantic role annotation, and then hires three annotators to carry out the annotation of Chinese semantic roles. Three annotators were hired to carry out the semantic role annotation. Annotator 1 was a graduate student majoring in linguistics, while annotators 2 and 3 were a master's degree and doctoral degree student majoring in computer science, respectively, and were given a one-week training prior to the annotation in order to carry out the semantic role annotation work in a better way.

During the annotation process, in order to facilitate the management and statistics, this paper divided 103871 Tibetan sentences into five groups evenly, and issued them to the three annotators in turn for annotation. When labeling, the consistency of the data labeled by different annotators is the most important index to measure the quality of labeling, and can also judge whether the specification is feasible, therefore, this paper adopts the Kappa coefficient to quantitatively measure the consistency of semantic role labeling.Kappa coefficient is a statistical measure of consistency, with the value of [-1,1], which means, respectively, namely, -1: no consistency at all, 0: accidental consistency, 0.0 ~0.20: very low agreement, 0.21-0.40: average agreement, 0.41-0.60: moderate agreement, 0.61-0.80: high agreement, 0.81-1: almost complete agreement. The Kappa coefficient based on the confusion matrix is calculated as follows:

$$kappa = \frac{p_o - p_e}{1 - p_e} \tag{40}$$

$$p_o = \frac{\sum_{i=1}^{28} a_{ii}}{N} \tag{41}$$

$$p_e = \frac{\sum_{i=1}^{28} a_{i+} \times a_{+i}}{N^2} \tag{42}$$

where p_o is the accuracy, i.e., labeling consistency, and p_e denotes chance consistency. The superscript 28 in Eq. (41) and Eq. (42) corresponds to the 28 semantic role labels contained in the specification, a_{ii} denotes the number of samples labeled consistently, N is the total number of samples, a_{+i} denotes the sum of the number of samples in the ith row, and a_{+i} denotes the sum of the number of samples in the ith column. The Kappa coefficients of specific each set of data labeling are shown in Fig. 5, where Legend A represents the result of labeling consistency verification between annotator 1 and annotator 2, Legend B represents the result of labeling consistency verification

between annotator 1 and annotator 3, and Legend C represents the result of labeling consistency verification between annotator 2 and annotator 3.

From the figure, it can be seen that for each group of data, the Kappa coefficient of the semantic role labels labeled between the three annotators can be maintained above 80%, and the average Kappa coefficient of all semantic role labels labeled is 82.43%, which indicates that the labeled data based on the specification developed in this paper has achieved better consistency, verified the feasibility of the developed specification, and achieved the expected goal.

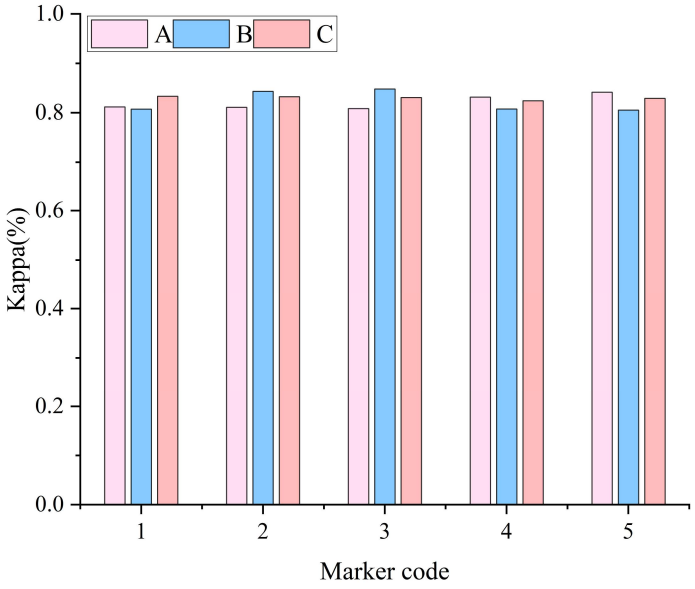

Fig. 5. Kappa coefficient

In order to more intuitively observe the distribution of each role label in the semantic role annotation dataset constructed in this paper, this paper counts the labeled frequency of each semantic role label in the semantic role library, which is shown in Fig. 6. It can be seen that the semantic roles of the giver, the party involved, and the recipient are labeled more often, while the semantic roles of the source, the starting point, and the end point are labeled less often, which is consistent with the actual linguistic phenomenon.

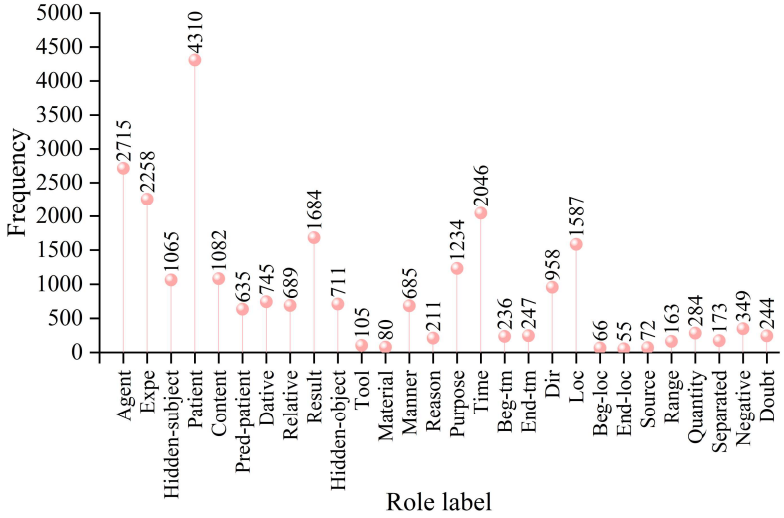

Fig. 6. The frequency of the tags of each semantic role

In addition, in order to verify the necessity of the new semantic roles in this paper, the distribution of the new semantic roles in the Tibetan semantic role annotation dataset was deliberately counted. The specific new roles and their distribution are shown in Table 5.

The total occurrence rate of new semantic roles is 13.44%, which occupies a certain proportion. It shows that the new semantic roles in this paper play a certain role in the process of semantic role labeling, which provides help in expressing the shallow semantic information of sentences.

Table 5. The distribution of new semantic roles in the data set

New semantic roles	Number of being labeled	Semantic role occurrence rate(%)
Hidden subject	1065	4.34
Hidden object	711	2.87
Predicate object	635	2.56
Starting time	241	0.97
Ending time	239	0.99
Sparable words	177	0.68
Question	258	1.03
Total emergence rate		13.44

5. Conclusion

In this paper, the dynamic assignment of semantics is based on principal component analysis, and the Highway Bi-LSTM model is established to improve the effect of Chinese semantic role labeling. The similarity distribution of this paper's algorithm is more concentrated, with higher correlation with manual scoring, and the correlation coefficient with the manual judgment value in the test dataset is more than 0.9. After adding the attention mechanism, the F1 value of the Bi-LSTM model on the two test sets improves by 18.56% and 8.95%, respectively, which proves the reasonableness of the design of this model. The F1 value of this model is higher than that of the comparison model, and it has better semantic role annotation performance with an average Kappa coefficient of 82.43%. Comprehensively, the method in this paper has a good effect of improving the role labeling of Chinese semantics.

References

[1] Song, H., Wang, S., Liu, Y., & Wang, Y. (2022). Predicate-attention neural model for Chinese semantic role labeling. Computers and Electrical Engineering, 99, 107741.

[2] Cai, R., & Lapata, M. (2020, November). Alignment-free cross-lingual semantic role labeling. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 3883-3894).

[3] San Kim, T., & Sohn, S. Y. (2020). Machine-learning-based deep semantic analysis approach for forecasting new technology convergence. Technological Forecasting and Social Change, 157, 120095.

[4] Salloum, S. A., Khan, R., & Shaalan, K. (2020). A survey of semantic analysis approaches. In Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020) (pp. 61-70). Springer International Publishing.

[5] Xu, N., & Mao, W. (2017, November). Multisentinet: A deep semantic network for multimodal sentiment analysis. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (pp. 2399-2402).

[6] Jaroli, P., Singla, C., Bhardwaj, V., & Mohapatra, S. K. (2022, April). Deep learning model based novel semantic analysis. In 2022 2nd international conference on advance computing and innovative

technologies in engineering (ICACITE) (pp. 1454-1458). IEEE.

[7] Jiang, B., & Lan, Y. (2019). Research on semantic role labeling method. In Communications and Networking: 13th EAI International Conference, ChinaCom 2018, Chengdu, China, October 23-25, 2018, Proceedings 13 (pp. 252-258). Springer International Publishing.

[8] Yang, H., & Zong, C. (2016). Learning generalized features for semantic role labeling. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 15(4), 1-16.

[9] Wang, Y., Wan, F., Ma, N., & Liu, D. (2019, June). Research on Chinese semantic role labeling with hierarchical syntactic clues. In 3rd International Conference on Economics and Management, Education, Humanities and Social Sciences (EMEHSS 2019) (pp. 190-196). Atlantis Press.

[10] Xu, K., Wu, H., Song, L., Zhang, H., Song, L., & Yu, D. (2021). Conversational semantic role labeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 2465-2475.

[11] Munir, K., Zhao, H., & Li, Z. (2021). Adaptive convolution for semantic role labeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 782-791.

[12] Liu, Y., Gao, J., Huang, H., & Liu, Y. (2023, March). Chinese Semantic Role Labeling Based on BILSTM-CRF Extended Model. In 2023 5th International Conference on Natural Language Processing (ICNLP) (pp. 182-186). IEEE.

[13] Tan, Z., Wang, M., Xie, J., Chen, Y., & Shi, X. (2018, April). Deep semantic role labeling with self-attention. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).

[14] Qian, F., Sha, L., Chang, B., Liu, L. C., & Zhang, M. (2017, September). Syntax aware LSTM model for semantic role labeling. In Proceedings of the 2nd Workshop on Structured Prediction for Natural Language Processing (pp. 27-32).

[15] Cai, R., & Lapata, M. (2019, November). Semi-supervised semantic role labeling with cross-view training. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 1018-1027).

[16] Zheng, X., Zhou, B., Huang, J., Liu, Y., & Wei, F. (2019, December). Chinese Semantic Role Labeling with Hybrid Model. In Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (pp. 462-467).

[17] Jin, G., Kawahara, D., & Kurohashi, S. (2018). Improving chinese semantic role labeling using high-quality surface and deep case frames. Journal of Natural Language Processing, 25(2), 201-221.

[18] Zheng, X., Chen, J., & Shang, G. (2017). Deep neural network-based Chinese semantic role labeling. ZTE Commun, 15, 58-64.

[19] Chen, H., Li, X., Zhang, M., & Zhang, M. (2024, August). Semantic Role Labeling from Chinese Speech via End-to-End Learning. In Findings of the Association for Computational Linguistics ACL 2024 (pp. 8898-8911).

[20] Zhu, A., Che, G., Wan, F., & Ma, N. (2021, January). Chinese semantic role labeling integrating global information and attention mechanism. In 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA) (pp. 184-188). IEEE.

[21] Wang, L., & Jiang, Y. (2024). Do translation universals exist at the syntactic-semantic level? A study using semantic role labeling and textual entailment analysis of English-Chinese translations. Humanities and Social Sciences Communications, 11(1), 1-12.

[22] Yuan, L. (2024). Semantic Role Labeling Based on Valence Structure and Deep Neural Network. IETE Journal of Research, 70(5), 5044-5052.

[23] Zhang Zhe,Chen Youling & Wang Xu. (2021). A semantic similarity computation method for virtual resources in cloud manufacturing environment based on information content. Journal of Manufacturing Systems646-660.

[24] Wenjie Li & Qiuxiang Xia. (2011). A Method of Concept Similarity Computation Based on Semantic Distance. Procedia Engineering(C),3854-3859.

[25] Óscar Alcón & Elena Lloret. (2015). Studying the influence of adding lexical-semantic knowledge to Principal Component Analysis technique for multilingual summarization. Linguamática(1),53-63.

[26] Xueyi Li,Kaiyu Su,Qiushi He,Xiangkai Wang & Zhijie Xie. (2023). Research on Fault Diagnosis of Highway Bi-LSTM Based on Attention Mechanism. Eksploatacja i Niezawodność – Maintenance and Reliability(2).