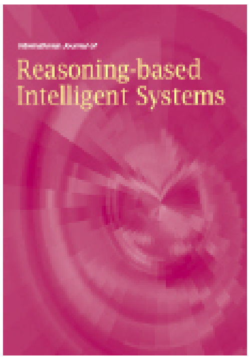
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Reasoning-based Intelligent Systems

ISSN online: 1755-0564 - ISSN print: 1755-0556
https://www.inderscience.com/ijris

Research on modified Biaffine method for Chinese semantic role labelling

Ning Ma, Jiahao Wang, Youqi Wang

DOI: 10.1504/IJRIS.2026.10076056

Article History:

Received:	03 November 2025
Last revised:	18 December 2025
Accepted:	18 December 2025
Published online:	17 February 2026

Research on modified Biaffine method for Chinese semantic role labelling

Ning Ma*, Jiahao Wang and Youqi Wang

Key Laboratory of Linguistic and Cultural Computing,
Ministry of Education,
Northwest Minzu University,
Lanzhou, 730030, China
and
Key Laboratory of China's Ethnic Languages
and Intelligent Processing of Gansu Province,
Northwest Minzu University,
Lanzhou, 730030, China
Email: maning8162@163.com
Email: 280166118@qq.com
Email: 2865295537@qq.com
*Corresponding author

Abstract: Semantic role labelling (SRL) is a core technology for semantic analysis. However, SRL methods based on pre-trained language models still face semantic ambiguity and training complexity. This paper proposes a Chinese SRL approach that integrates pre-trained language models with Biaffine technology to better capture semantic information in long sentences while alleviating training difficulty. By further incorporating pooling techniques and part-of-speech (POS) features, the model more accurately identifies semantic role boundaries. Experiments show that the RoBERTa-MPBF-CRF* model based on maximum pooling achieves an F1 score of 90.89% on the Chinese PropBank (CPB) dataset, outperforming CRF-only baselines. The introduction of POS features yields an average F1 improvement of about 1.5%, and the additional computational overhead remains acceptable relative to the performance gains.

Keywords: semantic role labelling; SRL; Chinese PropBank; pre-trained language models; Biaffine; pooling techniques; part-of-speech features; POS.

Reference to this paper should be made as follows: Ma, N., Wang, J. and Wang, Y. (2026) 'Research on modified Biaffine method for Chinese semantic role labelling', *Int. J. Reasoning-based Intelligent Systems*, Vol. 18, No. 8, pp.33–43.

Biographical notes: Ning Ma is a Professor at Northwest Minzu University and a Distinguished Feitian Scholar of Gansu Province. He is the Director of the Ministry of Education Key Laboratory of Language and Cultural Computing and leads the university's AI master's supervisor group. He serves on national committees for Chinese information processing and computational linguistics. His research focuses on machine learning and NLP, especially text generation, knowledge Q&A, and secure alignment for multimodal large models. He authored one monograph, published 60+ papers, led one NSFC and five provincial projects, holds three patents and six software copyrights, and won multiple awards.

Jiahao Wang graduated in 2024 from the China National Information Technology Research Institute, Northwest Minzu University, with an MS degree in Computer Technology. His research interests mainly include natural language processing and logical reasoning question answering.

Youqi Wang is first-year AI graduate student at Northwest Minzu University specialising in NLP alignment. He is proficient in Python and C, and experienced with PyTorch and TensorFlow, with strong hands-on capabilities in model training, debugging, and Linux-based deployment. His research outputs include one EI-indexed conference paper, a software copyright under review, and ongoing development of value-preference datasets. He previously worked as a Technical Engineer at Beijing Bochuang Shanghe Technology, where he delivered product training and provided technical support for university-level competitions. He is committed to advancing both the theory and practical applications of artificial intelligence.

1 Introduction

As a vital technology in natural language processing, semantic role labelling (SRL) mainly relies on pre-trained language models and CRF models for in-depth research. Nevertheless, these models often face challenges including numerous network parameters, complex training process, and insufficient processing of semantic information for long sentences. In order to more accurately capture semantic role boundary relationships, researchers have attempted to integrate linguistic features such as lexicality into the models Li et al. (2021), but this may also increase the complexity of model training. The Biaffine technique models the probability of semantic roles between words by constructing a Biaffine matrix, which centres on the use of a bilinear function to portray the interdependence between words Xu et al. (2022). Specifically, the technique combines vectors representing inter-word relationships with Biaffine matrices and realises nonlinear transformations through activation functions, while the parameters of the Biaffine matrices are learned and adapted during the model training process.

In the present study, we put forward a new method for Chinese semantic role annotation that combines pre-trained language models and Biaffine technology. First, the coding layer output of the pre-trained language model is efficiently sampled and extracted by pooling technique, which simplifies the network structure, effectively lowers the number of parameters required for model training, and shortens the training period. Meanwhile, the Biaffine technique is applied to construct a global dependency model of the lexical items within a sentence, which is able to comprehensively analyse the interactions between verbs and other lexical items, and thus significantly improves the accuracy of the prediction of semantic roles. Due to the parallel processing capability of Biaffine technology, the model in this study is also capable of efficiently handling large-scale semantic role annotation work. After a series of experimental validations, the approach proposed in this study can obviously improve the overall performance of the semantic role annotation model and enhance the capability of the pre-trained language model in capturing semantic role boundary attribution relationships.

To clarify the novelty of this work, our main contributions are summarised as follows:

1 We propose RoBERTa-based pooling-of-Biaffine architectures (APBF/MPBF) for Chinese SRL on CPB and, for the first time, systematically integrate maximum/average pooling, Biaffine tag prediction, part-of-speech (POS) features and CRF decoding within a unified framework. The best variant, RoBERTa-MPBF-CRF* (with POS), achieves an F1 score of 90.89% on CPB, significantly surpassing previous CPB baselines.

2 We conduct a comprehensive empirical study of how pooling techniques (min/avg/max), Biaffine vs. conditional random fields (CRF) tag prediction layers, and POS features jointly influence both model accuracy and computational efficiency.

3 We provide a detailed comparison with a strong BiLSTM-CRF baseline on CPB. Zhu et al. (2021a, 2021b) together with an analysis of training and prediction time, demonstrating that the proposed MPBF family substantially reduces the computational cost compared with RoBERTa-CRF while maintaining or improving performance.

2 Related research

Over the years, traditional statistical machine learning methods have been limited in their effectiveness in SRL. Meanwhile, with the increase of deep learning techniques, neural network-based approaches have indicated great potential in natural language processing Ranathunga et al. (2023). These neural network-based models use word embedding vectors to initialise the features of the input layer and adapt to specific tasks by adjusting hyperparameters. The architecture of deep learning is consisted of an input layer, an output layer, and multiple hidden layers, and the prediction of the final task is achieved by passing information between the layers and assigning different weights.

Blloshmi et al. (2021) propose an end-to-end SRL model (GSR) that jointly predicts predicates and arguments for both dependency-based and span-based formulations. Fei et al. (2021a) integrate constituency and dependency representations to better exploit syntactic information for SRL. Li et al. (2020) explore high-order SRL with enhanced structural features to capture richer interactions between predicates and arguments. For Chinese SRL, Wang et al. (2022) present a syntax-aware framework based on self-attention, while Fei et al. (2022) introduce an encoder-decoder based unified SRL architecture with label-aware syntax. Shi et al. (2020) reformulate SRL as syntactic dependency parsing, and Fei et al. (2021b) further develop end-to-end SRL via a neural transition-based model. Yuan (2022) incorporates valence information into a deep SRL model for Chinese; Li et al. (2023) study learning SRL from compatible label sequences; and Wang et al. (2020) propose a multi-cue Chinese SRL approach based on CRF. Zhou et al. (2022) cast end-to-end span SRL as word-based graph parsing. Zhu et al. (2021a, 2021b) propose Chinese SRL systems with attention and pooling-based feature grouping on CPB, which we adopt as our main CPB baselines.

In addition to neural models, reasoning-based intelligent systems provide an important perspective on semantic representation and inference. For example, Jain et al. (2021) argues that enriching logical semantics with an ontological structure reflecting commonsense knowledge can help address long-standing challenges in natural language semantics. Lu (2025) develops a semantic-based document management system that uses ontologies and reasoning engines to support knowledge discovery and complex queries. These studies demonstrate that explicit semantic modelling and reasoning can complement data-driven approaches, which is consistent with our goal of learning structured semantic role representations on top of pre-trained language models.

In contrast to Zhu et al.'s (2021a, 2021b), BiLSTM-CRF models that rely on hand-crafted argument features and pooling-based feature selection on top of recurrent encoders, our work builds directly on large pre-trained language models and studies how pooling techniques, Biaffine prediction and POS features can be combined within the RoBERTa encoder to improve both accuracy and efficiency on CPB. Previous Chinese SRL work with multi-cue or syntax-aware models Wang et al. (2022), Wang et al. (2020) mainly focuses on feature engineering or syntactic integration, whereas we emphasise the interaction between pre-trained representations, pooling-of-Biaffine architectures and linguistic features.

For Chinese SRL, a series of works have explored multi-cue and syntax-aware architectures, such as integrating valence information Yuan (2022), combining constituency and dependency syntax Fei et al. (2021a), or leveraging multi-task learning frameworks Yang et al. (2021). These models significantly improve Chinese SRL performance on various datasets. However, most of them focus on corpora such as Chinese FrameNet or task formulations that are not directly aligned with the CPB setting used in this paper. As a result, only Zhu et al.'s (2021b) BiLSTM-CRF systems, Zhu et al. (2021a) report CPB scores under a comparable experimental setup, which we therefore adopt as our main CPB baselines in Section 5.

3 Model construction

The fusion model RoBERTa-POBF consists of an input layer, a coding layer and a POBF layer. In the training phase, the model integrates a variety of pre-trained language models to effectively obtain the semantic associations between words within a sentence with the help of their in-depth understanding of the context. In the meanwhile, the model incorporates linguistic features, especially lexical annotation, a step that significantly improves the model's ability in defining core role boundaries and resolving contextual conflicts. Further, the model employs a pooling technique to filter the output of the encoding layer to extract key information and eliminate unnecessary redundancies, thus simplifying the training process. Ultimately, the Biaffine layer is responsible for modelling the global dependencies between words in a sentence. Figure 1 displays the overall architecture of the model.

3.1 Input layer

In this experiment, not only was a pre-trained language model employed to enhance the training effect of the SRL model, but the key linguistic feature of lexicality was also introduced. Lexical properties are categorised according to the roles played by words in grammatical structures, covering a wide range of lexical categories such as nouns, verbs, adjectives, etc. thus revealing the grammatical properties of words. When SRL is performed, nouns are usually associated with the role of either the giver or the receiver, while verbs are mostly labelled as the central role. By integrating lexical information into the experimental data, the model is capable of understanding more precisely the linguistic meaning represented by each label. The objective of SRL is to determine and label the full range of semantic roles in a sentence that are closely associated with a predicate. Lexical features provide valuable information for this purpose, which not only relate to the grammatical function of words in context, but also correlate to their semantic roles. By combining lexical and semantic role annotations, it is possible to accumulate a deeper understanding of sentence structure and clarify the relationship between each word and the role it plays. By integrating lexical information, the model is able to identify and localise the roles played by specific words in a sentence with higher accuracy, thus effectively eliminating uncertainty in the semantic role annotation process. This fusion strategy ensures high accuracy of the annotation results, and the partial lexical comparison table in Table 1 further demonstrates the advantages of combining lexical information.

Table 1 Partial part of speech comparison table

Lexical tag	*Hidden meaning*	*Lexical tag*	*Hidden meaning*
CC	Conjunctions	NN	Other nouns
JJ	Other noun modifiers	LC	Azimuths
VA	Predicate adjectives	PN	Pronouns
VV	Other verbs	AD	Adverbs
NR	Proper nouns	P	Prepositions
NT	Time nouns	AS	Action auxiliaries

In constructing the input layer, this model adopts a pre-trained language model for enhancing the understanding and capture of contextual information, which significantly improves the accuracy of recognising semantic links between words within a sentence. In addition, by incorporating lexical tagging, a key linguistic feature, the model demonstrates higher accuracy in delineating core role boundaries and is more efficient in handling contextual conflicts. By categorising words with detailed grammatical functions, such as nouns, verbs, adjectives, etc. the model further deepens the understanding of sentence structure. By adding lexical tags to the experimental dataset, the model is able to comprehend the linguistic meaning of each tag more deeply, which significantly improves the accuracy of semantic role prediction. As presented in Figure 2, the construction process of input layer feature mining is demonstrated.

Figure 1 Model structure (see online version for colours)

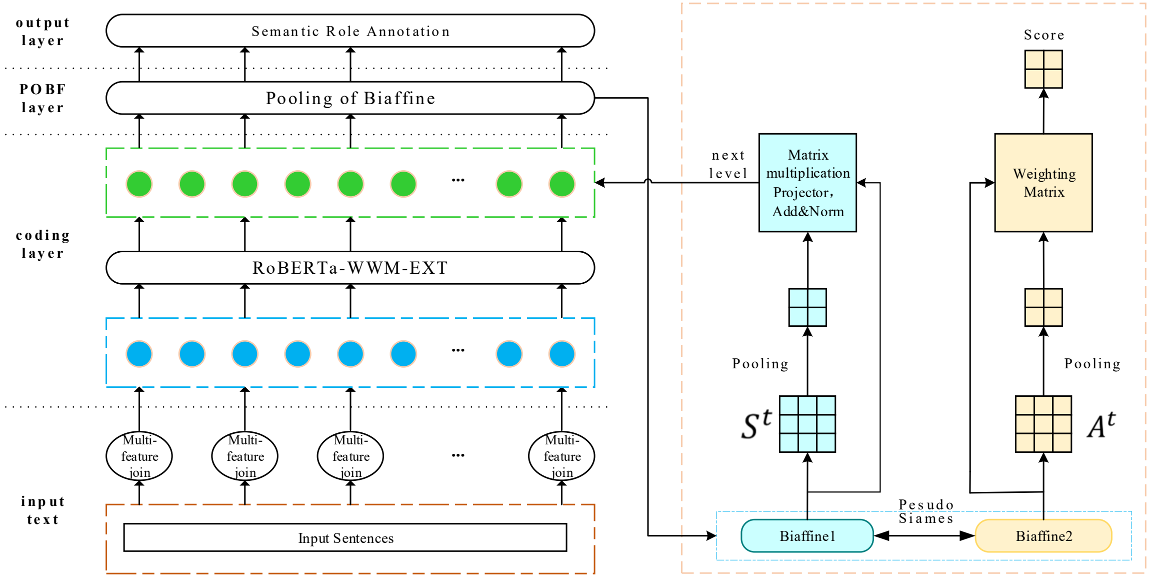


Figure 2 Input layer feature mining construction

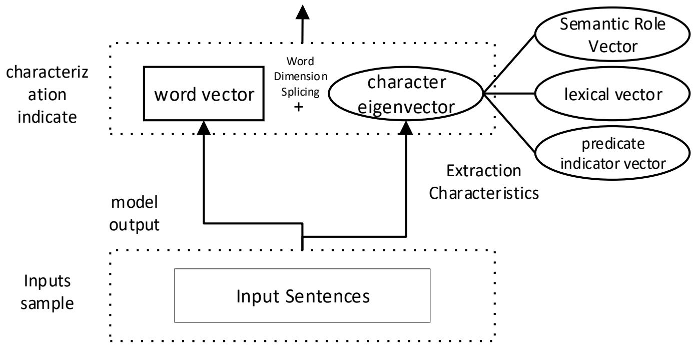
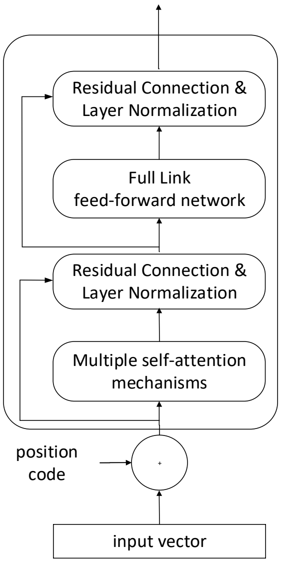


Figure 3 The overall structure of the encoder


Residual Connection & Layer Normalization
Full Link feed-forward network
Residual Connection & Layer Normalization
Multiple self-attention mechanisms
position code
input vector

3.2 Pre-trained language model coding layer

In the encoding stage of the model, three pre-trained language models, BERT, ALBERT and RoBERTa, are selected and subword-level encoding techniques, such as Word Piece or BPE methods, are utilised. Taking the RoBERTa model as an example, the architectural design of its encoder is shown in Figure 3, which clearly presents the overall structure of the model.

This paper employs diversified vector techniques to enhance the expressiveness of the input text. Specifically, they include the following:

1 Word embeddings, a technique that maps the words in the text into a low-dimensional space by means of a pre-trained word embedding method, which serves as the initialisation basis for the vocabulary list.

2 Segment embeddings, a vector that is used to mark the demarcation of different sentences in the input text. For the task of a single sentence, all input tokens are usually set to a uniform value, such as 0 or 1; for the task of sentence pairs, the segmental embeddings of two sentences are assigned different values.

3 Position embeddings, which are used to indicate the exact position of a word in a sequence and position information is integrated into the model by summing up with the word vectors.

4 Linguistic feature embeddings, which captures linguistic attributes in the text, such as lexical properties, to provide important clues to the model about sentence structure and grammatical roles, and helps to minimise misinterpretation of word roles.

5 Predicate indicator embeddings, which are special binary vectors used to explicitly mark the position of predicates in a sentence, where only the elements corresponding to the position of predicates are set to 1, and the rest of the positions are 0. This design allows the model to concentrate more on the semantic role recognition of predicates. The combined use of these vectors greatly enriches the model's understanding of the input text and enhances the accuracy and efficiency of SRL.

$$T=\left[x_1, x_2, x_3, ..., x_n\right] x_i E_i^{Token_{E}mb} E_i^{Pos_{E}mb} E_i^{Seg_{E}mb} E_i^{Lf_{E}mb} E_i^{Rel_{E}mb} \tag{1}$$

In model operations, all input tokens are concatenated into a unified sequence, serving as the input for the model to process downstream tasks. Regarding an input text sequence, where indicates the i^{th} character in text T, the sentence is split into word-level character sequences based on input layer features. These are then used to generate corresponding token embeddings through a pre-trained language model. Simultaneously, combined with positional vectors, segment vectors, linguistic feature vectors, and predicate indicators, they are concatenated into a new vector, which is then inputted to the next layer of the model. Ultimately, the comprehensive representation of these vectors forms the complete word embedding, as shown in (2).

$$E_i = \left[E_i^{TokenEmb}, E_i^{PosEmb}, E_i^{SegEmb}, E_i^{LfEmb}, E_i^{RelEmb}\right] \tag{2}$$

After concatenating the word vectors, the model feeds these vectors into a multi-head attention mechanism for further computation. This process involves first calculating three weight matrices: W_Q, W_K, W_V. Then these weight matrices are applied to each word vector, performing three linear transformations, resulting in new vectors q_t, k_t, v_t. Next, all the generated q_t vectors are concatenated into a large matrix to form the query matrix Q; similarly, k_t vectors are concatenated into the key matrix K, and v_t vectors are concatenated into the value matrix V. The construction and calculation of these matrices follow (3) to (5). In this way, the multi-head attention mechanism effectively captures information from different positions in the sequence and enhances the model's understanding of the text.

$$Q = Linear_q\left(E_i\right) = E_i W_Q \tag{3}$$

$$K = Linear_k\left(E_i\right) = E_i W_k \tag{4}$$

$$V = Linear_v\left(E_i\right) = E_i W_v \tag{5}$$

Calculate the attention mechanism matrix based on the query matrix, key matrix, and value matrix, as expressed in (6).

$$Att(Q, K, V) = softmax\left(\left(QK^T\right)\big/\sqrt{d_k}\right)V \tag{6}$$

In which, d_k refers the dimension of the key matrix K, K^T is the transpose of the key matrix K, the softmax function is a normalisation function, which multiplies the normalised matrix with the value matrix V, and finally obtains the attention matrix Att about Q, K, and V.

The multi-head attention mechanism consists of h attention heads, each with its own query, key, and value matrices Q_i, K_i, V_i. Through linear transformations, they are converted from a set (W_Q, W_K, W_V) to h sets $\left(W_0^Q, W_0^K, W_0^V\right), \left(W_1^Q, W_1^K, W_1^V\right), ..., \left(W_h^Q, W_h^K, W_h^V\right)$. Each attention head i computes attention over the input sequence, producing corresponding outputs $Head_i$. These outputs from all attention heads are concatenated and multiplied by an output weight matrix W^o, resulting in the final multi-head self-attention output MultiHead, as shown in (7) and (8).

$$Head_i = Att\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{7}$$

$$MultiHead\left(X_{input}\right) = concat\left[Head_i\right]W^O \tag{8}$$

After obtaining the attention matrix from the multi-head attention mechanism, it is supplemented to the input matrix for residual connection and layer normalisation, as shown in (9).

$$X_{hidden} = LayerNorm\left(X_{input} + MultiHead\left(X_{input}\right)\right) \tag{9}$$

In each layer of the encoder, there is a complete feed-forward network, which is consisted of two linear transformations with ReLU activation in between, as shown in (10).

$$FFN\left(X_{hidden}\right) = \max\left(0, X_{hidden}W_1 + b_1\right)W_2 + b_2 \tag{10}$$

Finally, calculate the residual connections and layer normalisation, as presented in (11).

$$X_{output} = LayerNorm\left(X_{hidden} + FFN\left(X_{hidden}\right)\right) \tag{11}$$

3.3 Pooling of Biaffine layer

In this model, the Biaffine layer precisely evaluates the interactions between words through a bilinear function, constructing a model of lexical dependency relationships. This layer receives processed hidden features from the preceding layer, which encapsulate crucial information about words in specific contexts. By computing attention scores between pairs of words, the Biaffine layer infers the semantic roles of each word and generates a distribution of role probabilities, assigning the most suitable semantic role labels to each word. The Biaffine layer excels in handling long-distance lexical dependency relationships, thanks to the combination of the bilinear function and nonlinear activation function, enabling the model to capture complex lexical relationships and significantly improve annotation accuracy and generalisation ability. In the training process, the parameters of this layer are optimised, endowing the model with strong expressive power and wide applicability, effectively improving the overall performance of the model in various SRL scenarios.

After processing at the encoding layer, the output sequence $X_{output} = \left[X_{output}^{0}, X_{output}^{1}, ..., X_{output}^{n}\right]^{T}$ is obtained. The Biaffine layer employs a dual-affine method to further process these outputs. As shown in Figure 1, Biaffine1 layer generates head vectors H_j and dependent vectors D_i through two multilayer perceptrons (MLPs) respectively, as detailed in (12) and (13). Subsequently, by calculating the attention scores between these two vectors, as shown in (14), the Biaffine layer can capture and understand the interrelationships between the words in the sentence more finely.

$$H_j = MLP_{head}\left(X_{output}\right) \tag{12}$$

$$D_i = MLP_{dep}\left(X_{output}\right) \tag{13}$$

$$S_{ij}^{T} = H_j Q\left(D_i\right)^{T} + H_j b \tag{14}$$

The input dimensions of MLP_{head} and MLP_{dep} are 2h, and the output dimension is d, where Q represents the learned parameters.

The obtained S_{ij}^{T} will undergo pooling techniques to derive its pooled results. As shown in (15), the result is the outcome of max pooling, while (16) represents the result of average pooling.

$$S_{ij}^{MaxP} = MaxP\left(S_{ij}^{T}\right) \tag{15}$$

$$S_{ij}^{AvgP} = AvgP\left(S_{ij}^{T}\right) \tag{16}$$

Finally, matrix Z is obtained through matrix multiplication and projection operations, as shown in (17). The generation of matrix Z involves multiplying the output matrix after pooling operation with the weight matrix P, followed by layer normalisation through *LayerNorm*, as shown in (18).

$$Z = S_{ij}^{Pooling} \times P \tag{17}$$

$$S_{ij}^{norm} = LayerNorm\left(S_{ij}^{T} + Z\right) \tag{18}$$

A_{ij}^{T} the Biaffine2 in Figure 1 shares the same structure as Biaffine1, but differs in parameter handling. The obtained through (11) to (16) is finally combined with the weight matrix α_t for prediction, as shown in (19).

$$A_{ij}^{Score} = \sum\nolimits_{t}^{T} \alpha_t \cdot softmax\left(A_{ij}^{T} \middle/ \sqrt{2h}\right) \tag{19}$$

The weight vector $\alpha_t \in R^{T}$ learned by the model satisfies the normalisation condition $\sum\nolimits_{t}^{T} \alpha_t = 1$.

For each position i, the decoder introduces the cross-entropy loss function to optimise predictions, with the specific form of the loss function detailed in (20) and (21).

$$loss = -\frac{1}{N}\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{C} y_{ij} \log\left(\hat{y}_{ij}\right) \tag{20}$$

$$\hat{y}_{ij} = A_{ij}^{Score} = \sum\nolimits_{t}^{T} \alpha_t \cdot softmax\left(A_{ij}^{T} \middle/ \sqrt{2h}\right) \tag{21}$$

where N refers to the number of samples, C is the number of classes, y_{ij} denotes the true label, and $\hat{y}_{ij}$ refers to the probability predicted by the model.

3.4 CRF labelling prediction layer

CRF is a type of probabilistic graphical model proposed by Lafferty et al. in 2001. The main advantage of CRF lies in its excellent predictive ability, capable of identifying and predicting label sequences that best match a given observation sequence. This model effectively captures the dependencies between observation sequences and label sequences by constructing the conditional probability distribution among labels. The details of label prediction computation in CRF can be understood through (22) and (23), which describe the computational process of the model in detail.

$$P(Y \mid X, \lambda) = \frac{1}{Z(X)}\exp\left(\sum\nolimits_{t}\sum\nolimits_{j}\lambda_j f_j\left(X, t, y_i, y_{i-1}\right)\right) \tag{22}$$

$$\text{where } Z(X) = \exp\left(\sum\nolimits_{y \in Y}\sum\nolimits_{i}^{n}\sum\nolimits_{j}\lambda_j f_j\left(X, t, y_i, y_{i-1}\right)\right) \tag{23}$$

One of the feature functions can be represented as $f(X, i, y_i, y_{i-1})$, where X denotes the input sentence, n is the current position, y_n is the current state, and y_{n-1} is the previous state. where $Z(X)$ denotes the normalisation constant, j denotes the number of feature functions, and λ_j is the weight of the feature function.

In the experiment of this study, the begin-inside-outside-end-single (BIOES) tagging scheme was adopted to enhance the recognition of semantic role boundaries and lexical attribution. This scheme subdivides the semantic roles in the sentence into multiple intervals, precisely locating each role and clearly defining the boundaries between roles. Utilising CRF technology, the model establishes a transition probability model between adjacent labels, enabling efficient learning and prediction of the optimal label sequence, achieving more precise semantic role annotation.

4 Experimental setup

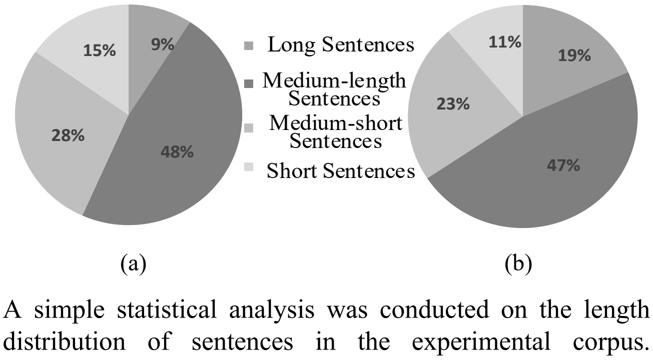
4.1 Experimental corpus

The CPB is a semantic role set developed specifically for Chinese corpora, defining a total of eighteen semantic roles. Among them, the predicate marker is labelled as ‘Rel’, and the core semantic roles ARG0-ARG5, as well as additional semantic roles ARG-M-X, are used for representation, where X is the abbreviation of the corresponding semantic role in English. For example, the marker for the temporal semantic role is labelled as ‘ARGM-TMP’. The detailed semantic roles are shown in Table 2.

For example, in the sentence ‘Zhang San ate an apple yesterday’, the predicate ‘ate’ is marked as Rel, ‘Zhang San’ is annotated as ARG0 (agent, the doer of the action), ‘apple’ is ARG1 (patient, the object being acted upon), and ‘yesterday’ is ARGM-TMP (temporal modifier). This example illustrates the SRL format of the CPB corpus.

The experiment utilised the CPB dataset, comprising 17,839 sentences in the training set and 1,115 sentences in the test set. Statistical analysis was conducted on the sentence length distribution in the experimental corpus. Figure 4 shows the proportion of different sentence lengths in the training and validation sets. Sentences with more than 100 characters are classified as long; those with 50–99 characters are medium-long; 30–49 characters are medium-short; and fewer than 30 characters are short. In the training set, medium-long sentences have the highest proportion at 48%, while long sentences have the lowest proportion at only 9%. In the validation set, medium-long sentences again have the highest proportion (47%), and short sentences the lowest (11%). These sentence length categories were determined based on the dataset’s distribution and linguistic intuition. Short sentences (fewer than 30 characters) usually contain a single clause, whereas long sentences (≥ 100 characters) often consist of multiple clauses and complex structures. This categorisation ensures that each group of sentences has sufficient examples and allows us to examine the model’s performance under different levels of sentence complexity.

Figure 4 Proportion of different sentence lengths in the CPB corpus, (a) training set (b) validation



(a) (b)

A simple statistical analysis was conducted on the length distribution of sentences in the experimental corpus. Figure 4 shows the proportions of different sentence types in the corpus. Sentences with more than 100 words are referred to as long sentences; those with fewer than 99 but more than 50 words are termed as medium-long sentences; those with fewer than 49 but more than 30 words are termed as medium-short sentences and those with fewer than 30 words are termed as short sentences. In the training set, it can be observed that medium-long sentences have the highest proportion, reaching 48%; while long sentences have the lowest proportion, accounting for only 9%. In the validation set, it can be found that medium-long sentences have the highest proportion, reaching 47%; and short sentences have the lowest proportion, accounting for only 11%.

4.2 Parameterisation

Table 3 lists the hyperparameter settings of our model. We chose these values based on preliminary experiments and the characteristics of the CPB dataset. Specifically, the maximum sequence length was set to 512 to accommodate the longest sentences, and the batch size was 4 due to GPU memory constraints with the large model. We used a learning rate of 1×10^{-5} for stable fine-tuning and applied a dropout rate of 0.1 in order to prevent overfitting. The model was trained for 70 epochs, which was sufficient for convergence without overfitting. The hidden layer size (768), number of hidden layers (12), and number of attention heads (12) follow the configuration of the RoBERTa base pre-trained language model.

4.3 Implementation details

All models were implemented in Python using the PyTorch framework and the HuggingFace transformers library. Training and evaluation were conducted on a single machine equipped with an Intel i7-6700HQ CPU, 8 GB RAM and an NVIDIA GTX 960M GPU (4 GB memory). The hyperparameters for all experiments follow the settings summarised in Table 3. For each configuration, we trained the model once with a fixed random seed to ensure reproducibility, and applied early stopping based on development-set F1 with a patience of 5 epochs. Owing to limited computational resources and the relatively large size of the RoBERTa-based encoders, we did not perform multiple independent runs per configuration; therefore, the reported results are single-run scores rather than averages over multiple runs.

5 Experimental

5.1 Experimental comparison

Experiment 1: evaluating the change in model performance after replacing the CRF tag prediction layer with the Biaffine tag prediction layer. As shown in Table 4, models marked with an asterisk (*) include POS features, while those without an asterisk are models without POS features. Training duration and prediction duration refer to the average time per epoch. This paper's test results adopt three common evaluation metrics: precision (P), recall (R), and F1-score (F1 value), defined as follows:

- Precision measures the proportion of correctly predicted semantic role labels in the SRL task.
- Recall measures the proportion of correctly predicted positives in the SRL task out of all true positives.
- F1-score: $F1 = (P \times R \times 2)/(P + R)$.

Analysis of Table 4 leads to the following conclusions:

1. The Biaffine model without integrated pooling techniques slightly underperforms the CRF model in terms of F1 score, with a difference of approximately 1%. However, the Biaffine model significantly reduces training and prediction times, with training time per epoch reduced by approximately 485 seconds and prediction time reduced by about 2 seconds.
2. Introducing pooling techniques before Biaffine technology improves model performance beyond models using only CRF technology, with an average increase in F1 score of around 2%. Additionally, Biaffine technology continues to demonstrate its advantage in reducing training and prediction times, with training time per epoch reduced by approximately 400 seconds and prediction time reduced by about 4 seconds.
3. The inclusion of integrated POS features in the Biaffine model leads to an average F1 score improvement of about 1.5%, indicating a positive impact of POS features on model performance. However, models incorporating POS features require more time for training and prediction, suggesting that the introduction of additional features increases computational burden. Nevertheless, given the performance improvement, this additional time cost is worthwhile.
4. The combination of pooling techniques with Biaffine technology significantly enhances model performance, particularly with average pooling and max pooling techniques. The average pooling technique increases the F1 score by approximately 5.5%, reduces training time by about 34 seconds, and reduces prediction time by about 1 second. The max pooling technique increases the F1 score by approximately 7%, reduces training time by about 44 seconds, and reduces prediction time by about 3 seconds. In contrast, the improvement effect of the min pooling technique is less significant, with an F1 score increase of approximately 1.5%.

Experiment 2: further validated the performance enhancement effects of max pooling and average pooling techniques on the model. Meanwhile, combined these two pooling techniques with Biaffine technology to construct the AvgPooling of Biaffine (APBF) model and MaxPooling of Biaffine (MPBF) model. The experimental results are shown in Table 5, where models with an asterisk (*) in their names include POS features, while those without an asterisk (*) do not. Through these experiments, the impact of different technology combinations on model performance can be further evaluated.

By comparing and analysing the data in Tables 4 and 5, the following conclusions can be made:

1. Integrating pooling techniques into the Biaffine feedforward neural network results in superior performance compared to models using pooling techniques solely in the Biaffine layer. Specifically, the F1 score of the RoBERTa-APBF model improved by approximately 2%, with a reduction in training time of about 127 seconds per epoch and a decrease in prediction time of around 7.5 seconds. Similarly, the RoBERTa-MPBF model exhibited an F1 score improvement of approximately 2.2%, with a training time reduction of about 134 seconds per epoch and a prediction time decrease of about 9 seconds. These results indicate that both the RoBERTa-MPBF and RoBERTa-APBF models significantly enhance the performance of SRL tasks.
2. Regarding training duration, models without a CRF layer, such as RoBERTa-APBF and RoBERTa-MPBF, have relatively shorter training times, implying an advantage in training efficiency. However, models with a CRF layer exhibit increased training times, which is reasonable considering the performance enhancement they provide. In terms of prediction time, all models maintain relatively low levels, indicating their ability to make fast predictions in practical applications. Particularly noteworthy is the RoBERTa-MPBF model, which achieves high F1 scores while requiring only 21.36 seconds for prediction, demonstrating excellent real-time prediction capabilities.
3. Models incorporating POS features show improvements in all performance metrics, indicating that POS information aids in a deeper understanding of sentence structure and semantic relationships. The RoBERTa-MPBF-CRF model achieves the highest F1 score of 90.89% among all models, validating the effectiveness of combining POS features with the MPBF layer in enhancing model performance.

Experiment 3: further investigation was conducted on the RoBERTa-MPBF model with the best performance to explore the influence of different convolutional kernel sizes on model performance. Testing was carried out by employing convolutional kernels of varying sizes in the pooling layer, aiming to understand whether the kernel size would affect the model's performance. Table 6 documents test results, with all convolutional kernels set to a stride of 1 and zero-padding of unit length applied at both ends of the input sequence. Through these experiments, a more comprehensive assessment of the specific effects of different technical parameters on model performance was achieved.

Table 2 CPB semantic role labels

Semantic role labeling	*Hidden meaning*	*Semantic role labeling*	*Hidden meaning*
ARG0	Agent (doer)	ARGM-FRQ	Frequency
ARG1	Patient (undergoer)	ARGM-LOC	Location
ARG2	Range	ARGM-MNR	Manner
ARG3	Action start	ARGM-ADV	Adverbial (general)
ARG4	Action end	ARGM-PRP	Purpose
ARG5	Other action-related	ARGM-BNF	Beneficiary
ARGM-DIS	Discourse marker	ARGM-TMP	Time
ARGM-DGR	Degree	ARGM-CND	Condition
ARGM-EXT	Extent	ARGM-TPC	Topic

Table 3 Hyperparameter settings

Parameter	*Description*	*Value*
Max_length	Maximum input sequence length	512
Batch_size	Number of samples per batch	4
Learning_rate	Learning rate	1e-5
Dropout	Dropout rate	0.1
Num_train_epochs	Number of training epochs	70
Hidden_size	Hidden layer dimension	768
Num_hidden_layers	Number of hidden layers (transformer blocks)	12
Num_attention_heads	Number of attention heads	12

Table 4 Performance comparison of various pooling techniques added before Biaffine technology

Models	*P/%*	*R/%*	*F1/%*	*Training duration/s*	*Projected duration/s*
RoBERTa-CRF	80.22	81.37	80.79	1,456.28	33.25
RoBERTa-CRF*	82.33	83.81	83.06	1,504.34	38.28
RoBERTa-Biaffine	78.34	80.74	79.52	971.28	31.98
RoBERTa-Biaffine*	79.18	83.09	81.09	1,015.21	33.41
RoBERTa-MinP-Biaffine	82.72	80.13	81.40	931.81	28.28
RoBERTa-MinP-Biaffine*	84.59	84.29	84.44	1,003.65	30.73
RoBERTa-AvgP-Biaffine	85.36	86.91	86.13	937.34	29.98
RoBERTa-AvgP-Biaffine*	87.92	88.24	88.08	1,009.47	32.11
RoBERTa-MaxP-Biaffine	86.88	86.42	86.65	927.17	28.47
RoBERTa-MaxP-Biaffine*	88.01	88.75	88.38	999.48	29.87

Note: Models marked with an asterisk (*) include POS features.

Table 5 Performance comparison of four models

Models	*P/%*	*R/%*	*F1/%*	*Training duration/s*	*Projected duration/s*
RoBERTa-APBF	87.62	88.65	88.13	810.02	22.41
RoBERTa-APBF*	88.98	90.48	89.72	883.49	24.69
RoBERTa-MPBF	89.83	87.77	88.79	793.22	21.36
RoBERTa-MPBF*	90.16	90.32	90.24	851.66	23.01
RoBERTa-APBF-CRF	89.27	90.11	89.69	1,452.74	35.57
RoBERTa-APBF-CRF*	89.97	90.40	90.19	1,533.32	38.15
RoBERTa-MPBF-CRF	91.51	86.48	88.92	1,430.96	33.88
RoBERTa-MPBF-CRF*	94.48	87.57	90.89	1,513.12	36.21

Note: Models marked with an asterisk (*) include POS features.

Table 6 Test results for different kernel sizes

Kernel size	*P/%*	*R/%*	*F1/%*	*Training duration/s*	*Projected duration/s*
2	89.79	87.49	88.62	792.91	21.18
3	89.83	87.77	88.79	793.22	21.36
4	91.71	71.53	80.37	810.69	24.05
5	90.30	56.51	69.51	837.46	37.73

On the basis of the experimental results in Table 6, the following conclusions can be drawn:

1 When the convolutional kernel size is 2 or 3, the model demonstrates higher stability, with the F1 score consistently between 88.6% and 88.8%. The training and prediction times of the model are relatively short, approximately 793 seconds and 21 seconds respectively, indicating that the model can maintain good performance while ensuring efficient training and prediction speeds.

2 As the convolutional kernel size increases to 4, the model's F1 score decreases to 80.37%, and the training and prediction times also increase to 810.69 seconds and 24.05 seconds respectively. This suggests that larger convolutional kernels may lead to a decrease in model performance and an increase in computational costs.

3 When the convolutional kernel size continues to increase to 5, the downward trend in model performance becomes more pronounced, with the F1 score dropping to 69.51%. At this point, the training and prediction times further extend to 837.46 seconds and 37.73 seconds respectively. This may indicate that excessively large convolutional kernel sizes can lead to overfitting or information loss, while also significantly increasing the demand for computational resources.

5.2 *Comparison with other experimental methods*

The final results of the text experiment are compared with other experimental methods, and the comparative results are presented in Table 7. Notably, Zhu et al. (2021b) achieved an F1 score of 81.41% on the CPB dataset with a BiLSTM-CRF model incorporating attention mechanisms and argument features, whereas our best model RoBERTa-MPBF-CRF* reaches 90.89%, demonstrating a significant improvement in performance.

To the best of our knowledge, Zhu et al. (2021a, 2021b), provide the only published CPB results that match our setting in terms of corpus, label set and evaluation protocol. Other recent Chinese SRL works mainly concentrate on different corpora or task formulations and do not report directly comparable CPB scores. For this reason, Table 7 focuses on Zhu et al.'s BiLSTM-CRF baseline as a representative CPB system, while Section 2 qualitatively discusses a broader range of Chinese SRL models.

Table 7 Comparison of final experimental results with other model results

Experimental methods	*F1/%*	*Training duration/s*	*Projected duration/s*
BiLSTM-Att-CRF-argument features	81.41	860.03	23.74
RoBERTa-MPBF-CRF* (ours + POS + CRF)	90.89	1,513.12	36.21
RoBERTa-MPBF* (ours + POS)	90.24	851.66	23.01

Note: Models marked with an asterisk (*) include POS features.

After analysing the predictive outcomes of the analysis model, it was found that the model has shortcomings in identifying semantic roles with lower occurrence frequencies including place names and personal names. To enhance the performance of the model in future research, exploration will be conducted to utilise large pre-trained language models such as ChatGLM, LLAMA, and LangChain. Through meticulous instruction adjustments, these models will be adapted and applied to SRL tasks, hoping to enhance the overall efficiency of the model.

6 Conclusions

To conclude, in the present study, a novel approach to Chinese SRL is proposed, which combines pre-trained language models with Biaffine technology. Experimental results indicate that this method obviously enhances the overall performance of SRL models and enhances the ability of pre-trained language models to capture semantic role boundary relationships. The impact of POS features, pooling techniques, and the combination of Biaffine and CRF technologies on model performance is also explored, along with some experimental results and conclusions. In future studies, further exploration will be conducted on fine-tuning with large pre-trained language models to further enhance model performance. These research findings enrich the study of SRL tasks and provide valuable references and insights for related research and applications.

One limitation of the current study is that, due to computational constraints, all results are based on single training runs per configuration; future work will include multi-run statistical analysis and more extensive comparisons with additional baselines.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62466052) and the Fundamental Research Funds for the Central Universities of Northwest Minzu University (Grant No. 31920250008).

Data availability

All relevant data are within the article and its Supplementary Materials. The CPB dataset used in this study is a standard Chinese SRL corpus. Due to license restrictions, the original CPB corpus itself cannot be redistributed by the authors, but it can be obtained from the original providers. To facilitate reproducibility, we release in the supplementary file S1 Data the processed CPB splits (training and development sets) in BIO and BIOES tagging formats, together with the BIOES SRL label set and the POS tag set used in our experiments.

Declarations

The authors declare no conflict of interests.

References

Blloshmi, R., Conia, S., Tripodi, R. et al. (2021) 'Generating senses and roles: an end-to-end model for dependency-and span-based semantic role labeling', *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, pp.3786–3793.

Fei, H., Li, F., Li, B. et al. (2022) 'Encoder-decoder based unified semantic role labeling with label-aware syntax', *Proceedings of the AAAI Conference on Artificial Intelligence*, pp.10822–10830.

Fei, H., Wu, S., Ren, Y. et al. (2021a) 'Better combine them together! Integrating syntactic constituency and dependency representations for semantic role labeling', *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp.549–559.

Fei, H., Zhang, M., Li, B. and Ji, D. (2021b) 'End-to-end semantic role labeling with neural transition-based model', *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, No. 14, pp.12803–12811.

Jain, S., Seeja, K.R. and Jindal, R. (2021) 'Computing semantic relatedness using latent semantic analysis and fuzzy formal concept analysis', *International Journal of Reasoning-based Intelligent Systems*, Vol. 13, No. 2, pp.92–100.

Lafferty, J., McCallum, A. and Pereira, F.C.N. (2001) 'Conditional random fields: probabilistic models for segmenting and labeling sequence data', *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pp.282–289.

Li, T., Kazeminejad, G., Brown, S.W. et al. (2023) 'Learning semantic role labeling from compatible label sequences', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 34, No. 5, pp.1501–1512.

Li, Z., Zhao, H., He, S. et al. (2021) 'Syntax role for neural semantic role labeling', *Computational Linguistics*, Vol. 47, No. 3, pp.529–574.

Li, Z., Zhao, H., Wang, R. et al. (2020) 'High-order semantic role labeling with enhanced structural features', *Proceedings of EMNLP*, pp.1134–1151.

Lu, C. (2025) 'Intelligent instructional resource management incorporating emotional and semantic features of user comments', *International Journal of Reasoning-based Intelligent Systems*, Vol. 17, No. 10, pp.10–19.

Ranathunga, S., Lee, E.S.A., Prifti Skenduli, M. et al. (2023) 'Neural machine translation for low-resource languages: a survey', *ACM Computing Surveys*, Vol. 55, No. 11, pp.1–37.

Shi, T., Malioutov, I. and Irsoy, O. (2020) 'Semantic role labeling as syntactic dependency parsing', *Proceedings of EMNLP*, pp.7551–7571.

Wang, X., Li, R., Wang, Z. et al. (2022) 'Syntax-aware Chinese framework semantic role labeling based on self-attention', *Journal of Chinese Information Processing*, Vol. 36, No. 10, pp.38–44.

Wang, Y., Wan, F. and Ma, N. (2020) 'Multi-cue Chinese semantic role labeling based on conditional random fields', *Journal of Yunnan University (Natural Sciences)*, Vol. 42, No. 3, pp.474–480.

Xu, Z., Wang, H. and Wang, B. (2022) 'Multi-Layer pseudo-Siamese Biaffine model for dependency parsing', *Proceedings of the 29th International Conference on Computational Linguistics*, pp.5476–5487.

Yang, J., Sun, J., Wang, Y. and Wang, Y. (2021) 'A multi-task learning framework for Chinese semantic role labeling', *Neurocomputing*, Vol. 429, pp.36–45.

Yuan, L. (2022) 'Semantic role labeling based on deep neural network fusing valence information', *Mini-Micro Systems*, Vol. 43, No. 9, pp.1925–1930.

Zhou, S., Xia, Q., Li, Z. et al. (2022) 'Fast and accurate end-to-end span-based semantic role labeling as word-based graph parsing', *Computational Linguistics*, Vol. 48, No. 2, pp.4160–4171.

Zhu, A., Wan, F. and Ma, N. (2021a) 'Chinese semantic role labeling by integrating pooling techniques and feature grouping', *Journal of Yunnan University*, Vol. 43, No. 5, pp.906–912.

Zhu, A., Wan, F., Ma, N. et al. (2021b) 'Multi-strategy Chinese semantic role labeling incorporating attention mechanisms', *Journal of Xiamen University*, Vol. 60, No. 6, pp.1019–1023.