



# Relation extraction: advancements through deep learning and entity-related features

Youwen Zhao<sup>1</sup> · Xiangbo Yuan<sup>1</sup> · Ye Yuan<sup>1</sup> · Shaoxiong Deng<sup>1</sup> · Jun Quan<sup>1</sup>

Received: 7 April 2023 / Revised: 4 May 2023 / Accepted: 6 May 2023 / Published online: 10 June 2023  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Austria, part of Springer Nature 2023

## Abstract

Capturing semantics and structure surrounding the target entity pair is crucial for relation extraction. The task is challenging due to the limited semantic elements and structural features of the target entity pair within a sentence. To tackle this problem, this paper introduces an approach that fuses entity-related features under convolutional neural networks and graph convolution neural networks. Our approach combines the unit features of the target entity pair to generate corresponding fusion features and applies the deep learning framework to extract high-order abstract features for relation extraction. Experimental results from three public datasets (ACE05 English, ACE05 Chinese, and SanWen) indicate that the proposed approach achieves F1-scores of 77.70%, 90.12%, and 68.84%, respectively, highlighting its effectiveness and robustness. This paper provides a comprehensive description of the approach and experimental results.

**Keywords** Relation extraction · Fusion feature · Semantic structure · Graph convolution

## 1 Introduction

Relation extraction is a crucial task in natural language processing that aims to automatically identify the semantic relation between a target entity pair in a sentence, forming a structured triad of “ $e_1, r, e_2$ ”. After years of development, the results of relation extraction have made significant progress and have been successfully applied in various fields, such as question answer (Ojokoh et al. 2023), knowledge base construction (Niu et al. 2012), and medical analysis (Matthews et al. 1990). However, there are still challenges to be overcome (Ranjan et al. 2022).

One of the main challenges is the inaccurate recognition of the target entity pair. In many relation instances, multiple entity pairs are available, but the structure of entity pairs is complex. For example, in the sentence “The ferocity of COVID-19 has affected the physical health of many people”, a relation extraction model will identify the semantic relation between the entities “COVID-19” and “health” as “Impact”, as shown in Fig. 1. However, the entities “health” and “physical health” are considered as two different entities, which may be used in different instances for relation extraction. Therefore, it is necessary to design a relation extraction method that is more focused on the target entity pair.

Another challenge is the feature sparsity problem brought about by the limited words in a sentence. The input of the relation extraction model is usually a raw sentence containing the target entity pair, but the entity-related features contained in short sentences are limited, resulting in a feature sparsity problem.

To face these challenges, we propose our approach to fusing entity-related features based on the deep learning framework. This approach first extracts unit features of the target entity pair from a sentence. In relation extraction, the model aims to capture contextual information of the entity pair from sentences, which is helpful for recognizing entity relationships. However, since sentences containing entity

✉ Youwen Zhao  
ywz\_moutai@163.com

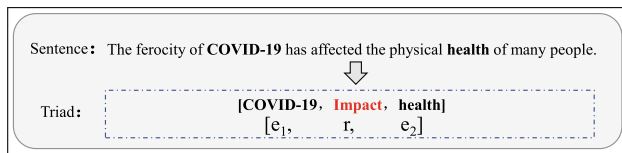
Xiangbo Yuan  
279894620@qq.com

Ye Yuan  
10323975@qq.com

Shaoxiong Deng  
20213803@mountain.com.cn

Jun Quan  
simon\_quan@mountain.com.cn

<sup>1</sup> Digital and information management center, Kweichow Moutai Co., Ltd, Zunyi 564501, Guizhou, China



**Fig. 1** A relation extraction example

pairs are usually short, they may lack sufficient semantic information, leading to the problem of feature sparsity. This sparsity hinders the model's ability to achieve high-precision entity semantic relationship recognition. Therefore, this paper proposes to enrich the model's semantic information by using features related to the target entity. These features are derived from datasets (manual labeling) and third-party tools (such as NLTK and Jieba). We define unit features as the basic semantic and syntactic properties of the target entity pair and its surrounding context, which help in addressing the feature sparsity problem. Examples of unit features include entity type, entity subtype, part of speech of entity word, and other features. A detailed introduction to unit features will be provided in Sect. 3. The model then generates fusion features based on man-made combination rules, enhancing its ability to sense the target entity pair. Next, the conventional convolution kernels and the graph convolutional kernels are adopted to extract high-order abstract representations of the semantics within the sentence. Finally, it predicts the most likely relation type from a predefined relation set through a linear classification layer.

The contributions of this paper are as follows:

- (1) We design a strategy to extract and combine entity-related unit features of the target entity pair from the sentence, which helps address the feature sparsity problem in relation extraction.
- (2) We propose a deep learning framework that integrates both conventional convolution kernels and graph convolutional kernels, which effectively extracts semantics and structure in a sentence based on entity-related unit features, addressing the challenges of inaccurate entity pair identification and feature sparsity in relation extraction.
- (3) We demonstrate the effectiveness of our proposed approach on three public datasets in Chinese and English, specific and comprehensive fields respectively, achieving F1-scores of 77.70%, 90.12%, and 68.84% on the ACE05 English, ACE05 Chinese, and SanWen datasets.

The remainder of this paper is organized as follows: Sect. 1 provides the background and motivation for our proposed model; Sect. 2 examines related works in relation extraction; Sect. 3 introduces the concept of unit features and their

extraction and use in this model; Sect. 4 outlines the methodology design and implementation details of our proposed approach; Sect. 5 presents experimental results, analysis, and comparisons to related works.

## 2 Related works

Relation extraction is an important task in natural language processing that has been widely studied (Wang et al. 2022). In this section, we categorize the related works into three groups: early rule-based methods, machine learning-based methods, and deep learning-based methods.

Early rule-based methods rely on syntactic analysis and grammar rules to define relations between entities (Zhang et al. 2009). These methods offer strong interpretability and high accuracy but are limited by the need for manually constructed rules and their heavy dependence on artificial knowledge bases.

Machine learning-based methods have become the mainstream approach in relation extraction (Kumar 2017; Waheeb et al. 2020). These models utilize statistical language models to automatically learn entity features from labeled data, offering strong generalization ability, adaptability, and scalability. However, they suffer from low interpretability, high dependence on labeled data, and a limited ability to capture complex features.

Deep learning-based methods have been shown to outperform traditional machine learning-based methods in relation extraction. These models leverage deep neural networks to learn high-order abstract features and representations from sentences. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been widely used in relation extraction. CNN-based models are effective in capturing local features of sentences but are difficult to capture long-distance dependencies (Liu et al. 2013). To address this limitation, some researchers have combined RNN with entity position indicators to obtain long-distance dependencies and improve performance (Zhang and Wang 2015). Other models, such as convolutional RNN (CRNN) (Song et al. 2019) and graph convolutional networks (GCNs) (Hu et al. 2022; Zhu et al. 2019), have been proposed to capture multi-level features and global dependencies. They also can benefit from pre-training models (Devlin et al. 2018; Peters et al. 2018). But models based on deep learning still have difficulties in capturing the semantics and structure of the target entity due to the raw sentence input.

In recent years, various deep learning-based models have been proposed to enhance the recognition ability of entity relations, including:

- (1) *Boundary modeling and enhancing* Several prior explorations focus on modeling and enhancing boundaries

for entity and relation extraction (Fei et al. 2020; Wang and Lu 2020; Fei et al. 2021, 2022). These works include constructing entity graphs, designing table-sequence encoders, employing pointer networks for discontinuous NER, and developing structure-aware generative language models.

- (2) *Joint entity and relation extraction* A number of works concentrate on joint entity and relation extraction using methods such as pointer networks (Fei et al. 2021), generative models (Fei et al. 2022), and table-filling methods (Wang and Lu 2020; Li et al. 2022).
- (3) *Employing external knowledge* Some existing works enhance boundary detection by employing external knowledge (Wu et al. 2021; Fei et al. 2021).
- (4) *Graph-based semantic structure modeling* There are also studies that focus on using graph-based approaches for semantic structure modeling in information extraction or related tasks (Fei et al. 2022, 2020, 2020).

In summary, while earlier rule-based methods are highly interpretable and accurate, they have limitations due to their dependence on manually constructed rules and artificial knowledge bases. Machine learning-based methods have the advantage of strong generalization ability and scalability, but they suffer from low interpretability and dependence on labeled data. Deep learning-based methods have shown superior performance in relation extraction, but the lack of interpretability is still a concern. In this paper, we propose an approach to extract high-order abstract features by combining entity-related features based on convolutional and graph convolution neural networks. Our proposed approach aims to overcome the feature sparsity problem and focus on the target entity pair in relation extraction, improving its performance. Next, we introduce the extraction and combination of unit features related to the entity pair in this model.

### 3 Entity-related features

In relation extraction, the target entity pair plays a crucial role in identifying the semantic relation between them. Thus, capturing the entities' semantics and structure is essential. This section presents an overview of the formal representation of relation extraction and discusses the entity-related features.

Let  $S = [x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_u, \dots, x_v, \dots, x_N]$  be a sentence with  $N$  words. In the sentence, the target entity pair is represented by  $e_1 = [x_i, \dots, x_j]$  and  $e_2 = [x_u, \dots, x_v]$ . In relation extraction, each relation mention is assigned a predefined relation type from a set of  $R = [r_0, r_1, r_2, \dots, r_L]$ , which contains  $L + 1$  predefined relation types. Here,  $r_0$

**Table 1** Unit features

Representation	Feature name	Meaning
$U_1$	Type	Type of entity
$U_2$	$L_{\text{POS}}$	Part of speech of the left word
$U_3$	$R_{\text{POS}}$	Part of speech of the right word
$U_4$	$E_{\text{Position}}$	Structure of the entity pair
$U_5$	Marker	Markers of an entity

In this table, we introduce the five unit features proposed in this paper. These features are related to the target entity pair and can express the semantic information of the entities. The meaning of each unit feature is explained

represents a negative relation type, while  $r_1$  to  $r_T$  represent  $T$  positive relation types in the dataset.<sup>1</sup>

In relation extraction, the target entity pair is the central part of the sentence. Generally, a classifier model may easily neglect the structure and semantics of the target entity pair if the raw sentence is input into the model. To capture more features about the target entity pair from a sentence, we first extract unit features related to the entity pair and then combine them using man-made rules to generate fusion features.

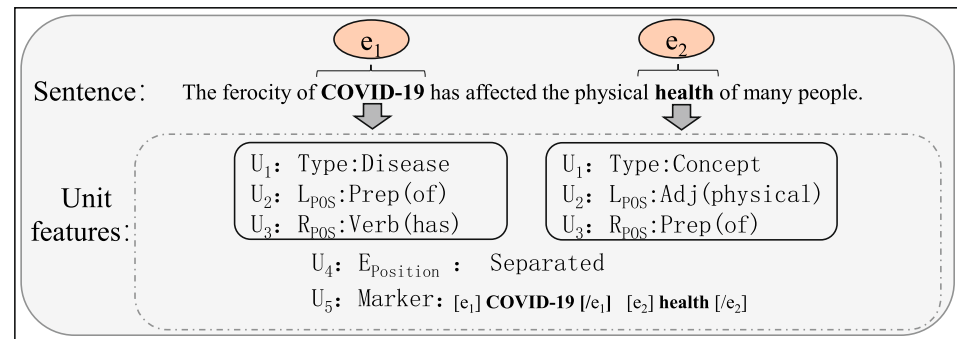
We propose two ways to generate unit features about the entity pair. One way is to extract the annotated features by domain experts within the dataset. The other way is to acquire additional features using third-party tools. All entity-related features obtained by these two methods are primarily annotated with respect to the surrounding context, enabling the model to learn the semantics and structure of the target entity pair.

To finalize our feature selection, we systematically integrated the dataset annotations, third-party tools, and manual selection methods. Through this rigorous process, we identified five crucial unit features for feature combination. A comprehensive representation of these features is provided in Table 1.

The first unit feature, entity type, effectively conveys the semantics of entities, encompassing categories such as "person", "location", "organization" and others. This feature is derived from dataset annotations and is represented as  $U_1$  in Table 1. The second and third unit features associated with entity semantics are the part-of-speech (POS) of the words adjacent to the entity, such as "n" (noun), "v" (verb), and "prep" (preposition), obtained through third-party tools (JieBa or NLTK) and are denoted as  $U_2$  and  $U_3$  in Table 1.

Our analysis shows that nested named entities account for 29.80% of the ACE05 Chinese dataset, 22.76% of the English dataset, and 11.27% of the SanWen dataset. Identifying

<sup>1</sup> In this study, the SanWen dataset does not include negative instances.

**Fig. 2** Unit features in a relation instance

the structure of entity pairs with nested entities is crucial in relation extraction, as these structures play a vital role in recognizing entity relations. To address this issue, we propose a new feature,  $U_4$ , which describes the structure of the target entity pair as either "nested" or "separated" types. The former implies that  $e_1$  is within  $e_2$  or vice versa, while the latter indicates that  $e_1$  is before  $e_2$  or vice versa. The relative structural features between two entities contained in the target entity pair play an important role in identifying their semantic relation.

To ensure that the recognition result remains unaffected by other entities, it is critical to obtain the semantics, position, and structure of the target entity pair. Therefore, we introduce the unit feature  $U_5$ , which incorporates the entity type and the structure of the entity pair into the entity marker while distinguishing their boundaries. This enables the model to learn rich features of the target entity pair.

For example, in the sentence shown in Fig. 1, the five unit features mentioned in Table 1 of the target entity pair "COVID-19" and "health" are shown in Fig. 2.

The unit features of the entity pair encapsulate the individual entity semantics, allowing the model to comprehend the contextual significance of the entity. However, relying solely on these features may overlook the intricate interconnections among them, thereby impeding the recognition of semantic relations between entity pairs.

To address this issue, our study introduces seven fusion features by combining unit features. The fusion features empower the model to learn long-distance semantics, enrich the features not annotated within the dataset, and capture the association information between target entities. We thoroughly illustrate the carefully devised combination rules for these fusion features in Table 2.

The individual semantics of each entity within an entity pair are encapsulated by the unit features, enabling the model to understand its contextual importance. However, solely relying on these unit features can cause the model to overlook the intricate interconnections among the entities, ultimately hindering the recognition of semantic relations between the pairs. Therefore, we introduce seven fusion features, which are derived by combining the unit features.

**Table 2** Fusion features

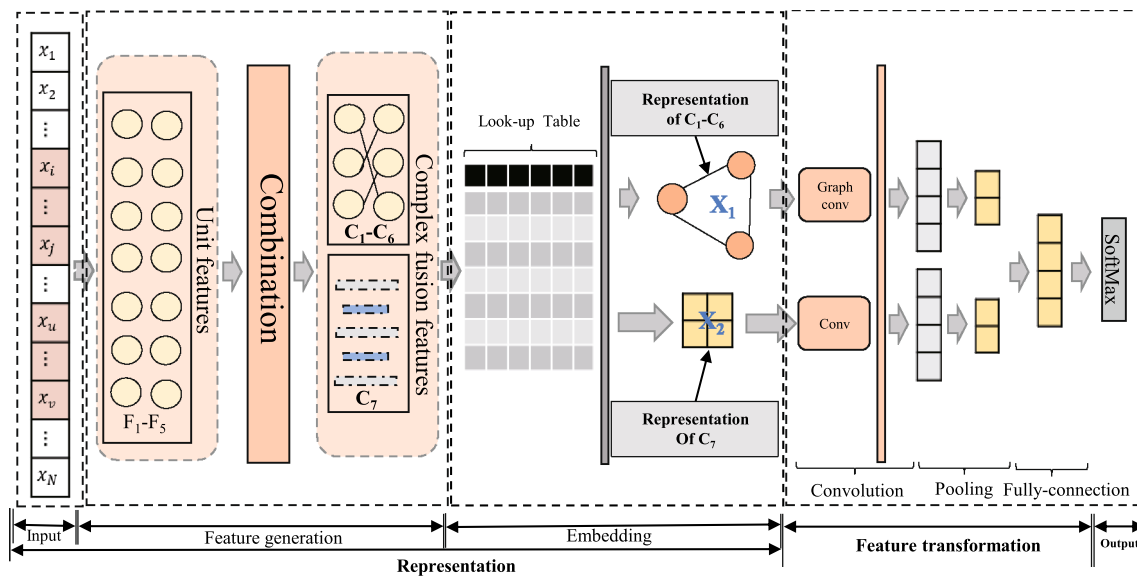
Representation	Feature name	Meaning
$F_1$	$Type(e_1) \oplus Type(e_2)$	Union of entity types
$F_2$	$L_{POS}(e_1) \oplus L_{POS}(e_2)$	Union of left POS for entity pair
$F_3$	$R_{POS}(e_1) \oplus R_{POS}(e_2)$	Union of right POS for entity pair
$F_4$	$L_{POS}(e_1) \oplus R_{POS}(e_1)$	Union of left and right POS of $e_1$
$F_5$	$L_{POS}(e_2) \oplus R_{POS}(e_2)$	Union of left and right POS of $e_2$
$F_6$	$E_{Structure}$	Entity pair structure
$F_7$	Sentence & Marker	Sentence with entity markers

This table illustrates the fusion features used in our research, with each feature representing an aspect of the entity pair's information or structure

These fusion features enhance the model's ability to learn long-distance semantics, supplement the features that are not annotated in the dataset, and capture the association information between target entities. We provide a thorough illustration of the carefully devised combination rules for these fusion features in Table 2.

Table 2 presents seven fusion features, where  $F_1 - F_5$  are derived by concatenating two unit features using the "-" connector.  $F_1$  combines the " $U_1$ " feature of both target entities, providing a comprehensive representation of entity semantics by reflecting their categorical affiliation. Target entity types are crucial in distinguishing entity relations. For example, a relation involving two "person" entities is unlikely to represent a "location" and more plausibly signifies "social" associations.  $F_2 - F_5$  encompass the amalgamation of adjacent word POS features, highlighting the semantic connections between target entities and their context.  $F_6$  denotes a composite feature illustrating the structure between target entities, while  $F_7$  represents a semantic sequence generated by inserting entity markers containing entity-related features on both sides of target entities within a sentence.

The utilization of fusion features allows for the integration of both internal and external semantics of target entities. The extraction of high-order abstract features from fusion features allows the model to learn the fusion information of



**Fig. 3** Framework of our proposed model

entity-related features. While the fusion of the unit features can express the surface-level association between features, it is essential to capture their deeper connections through high-order abstract features, which can be extracted using a deep learning model. In this paper, we propose a novel deep learning network architecture that focuses on the fusion of features.

## 4 Our proposed model

Our proposed model is structured by three modules: “representation”, “feature transformation”, and “output”. The overall framework of the model is depicted in Fig. 3.

### 4.1 Representation

The representation module consists of three components: input, feature generation, and embedding, which transform the input sequence into a vector matrix. The input sentence is denoted as  $S = [x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_u, \dots, x_v, \dots, x_N]$ , where the entity pair  $e_1 = [x_i, \dots, x_j]$  and  $e_2 = [x_u, \dots, x_v]$  (with  $1 \leq i \leq j \leq u \leq v \leq N$ ) represents the two target entities, as illustrated in “input” component.

The “feature generation” component facilitates feature extraction and combination operations. Initially, the model derives a set of unit features from the input sentence,  $U_1 - U_5$ , delineated in Table 1. To better capture the semantics in the sentence, fusion features  $F_1$  to  $F_7$  are constructed based on the connections among unit features, as depicted in Table 2. The combination operation for unit features is shown in Fig. 3’s “combination” exhibit. For example,  $F_1$

combines entity type features of  $e_1$  and  $e_2$ . Given two unit features “Per” and “Loc”, a fusion feature “Per\_Loc” is generated by a connection operation, which offers an improved representation of the semantic correlation between entity features. Consequently, the neural network model can learn the semantic structure inherent to the target entity pair. Fusion feature  $F_i$  is formulated from two unit features  $U_j$  and  $U_k$  and can be expressed as:

$$F_i = U_j \oplus U_k \quad (1)$$

The feature generation operation transforms the input sentence  $S$  containing  $N$  words into a sequence of fusion features  $\hat{S} = [F_1, F_2, \dots, F_7]$ . The sequence then is transformed into the embedding layer. The objective of the embedding layer is to convert fusion features in the sentence into a low-dimensional dense matrix via the word vector lookup table  $W_e$ . This approach offers reduced computational complexity and text similarity calculation capabilities compared to traditional one-hot vector representations, as outlined in (Church 2017). Initializing  $W_e$  in the embedding layer is essential for obtaining the desired text semantic representation. In this experiment, pre-trained word vector files Google-News and Wiki-100 were employed for initialization.

The representation module’s final output is the representation of the input sentence, segmented into representations of  $F_1 \sim F_6$  and  $F_7$ , where  $F_7$  is the original sentence embedded the entity marker  $U_5$ . Each fusion feature, except  $F_7$ , is represented by a feature vector, while each word in  $F_7$  is represented by a vector through the embedding layer in the representation module. The vector matrix for all fusion features is denoted as  $X = [x_1, x_2, \dots, x_L]$ , where  $v_i \in R^d$  represents



a feature vector. The vector dimension ( $D$ ) is set to 300 for Google-News in English and 100 for Wiki-100 in Chinese, based on the pre-trained word vector used in the experiment. The vocabulary size of a dataset employed in the experiment is represented by  $Q$ , and the word vector matrix is expressed as  $W_e = R^{D \times Q}$ . When mapping the input sentence to the model, the sentence length must be fixed. If the length exceeds the cropping value, it will be automatically padded.  $L$  denotes the input length of each instance text. The output process of the embedding layer can be described as:

$$X = \text{embedding}(\hat{S}) = [x_1, x_2, \dots, x_L] \quad (2)$$

The sentence representation  $X$  obtained from the embedding layer can be divided into two parts:  $X_1 = [x_1, x_2, \dots, x_k]$  represents features  $\mathbf{F}_1$  to  $\mathbf{F}_6$ , and  $X_2 = [x_{k+1}, \dots, x_L]$  represents feature  $\mathbf{F}_7$ .

## 4.2 Feature transformation

In the feature transformation module, high-order abstract features are extracted from  $X_1$  and  $X_2$ , output by the upper layer, using two different extractors.

For  $X_1$ , we employ graph convolution to capture the global dependencies between fusion features. Specifically, the input sentence is represented as a set  $\langle E, N \rangle$  consisting of edges ( $E$ ) and nodes ( $N$ ). Then, a feature connection matrix  $M$  is designed to establish connections between fusion features. The elements in  $M$  are randomly assigned either 0 or 1 to clean or build connections for fusion features at corresponding locations. We then apply the graph convolution operation on  $X_1$  and  $M$ . Weight matrix  $W$  and bias value  $b$  are generated by using a truncated normal distribution for model learning during network training. The graph convolution operation can be expressed as follows:

$$H^G = \text{GraphConv}(X_1, M, W, b) = \sigma(X_1 \cdot M \cdot W + b) \quad (3)$$

Here,  $\sigma$  denotes a nonlinear activation function.

For  $X_2$ , which represents the tagged sentence by the entity marker  $U_5$ , we employ conventional convolution kernels with windows of multiple sizes to capture the structure and semantics of the target entity pair in the sentence. The convolution operation can be described as follows:

$$H^C = \text{Conv}(X_2, W_c, b) = \sigma(W_c \cdot X_2 + b) \quad (4)$$

Here,  $W_c$  is the convolution window with a size of  $K \times D$ , where  $K$  denotes the number of vertically spanning words ( $K \in [2, 3, 4, 5]$ ), and  $D$  corresponds to the word vector dimension.  $b$  is the bias value, and  $\text{Conv}$  denotes the convolution operation.

The graph convolution and convolution operations on the representation of the sequence of fusion features effectively captures local features and the interconnection of all features. It maps the vector matrix  $X_1$  and  $X_2$  to high-order feature representations, enabling the learning of  $K$ -degree semantics from the  $K \times D$  convolution window.

To extract effective classification features and filter out unwanted features, maximum pooling operations are performed on  $H^G$  and  $H^C$ , respectively. This process can be formally expressed as:

$$p_1 = \text{maxpool}(H^G) \quad (5)$$

$$p_2 = \text{maxpool}(H^C) \quad (6)$$

Finally, we concatenate the two pooled outputs and apply a fully connected operation (fc) to generate the final output. This operation can be represented as follows:

$$p = \text{fc}([p_1, p_2]) \quad (7)$$

## 4.3 Output

To obtain probability values for each type and make the final prediction, we employ a softmax function as the output module:

$$y = \text{softmax}(p) \quad (8)$$

In summary, our proposed model extracts the high-order representation of fusion features by performing two operations: graph convolution and convolution with multiple kernels when a sentence  $S$  is input. Specifically, in Eq. 9, we use  $f_1$  and  $f_2$  to represent the two convolution operations, and  $\mathbf{F}_f$  and  $\mathbf{F}_l$  represent the first six and the last fusion features, respectively. The complete process can be summarized by the following equation:

$$y = \text{Softmax}(\text{fc}([\text{maxpool}(f_1(\text{embedding}(\mathbf{F}_f))), \text{maxpool}(f_2(\text{embedding}(\mathbf{F}_l))))]) \quad (9)$$

## 5 Experiment

In this Experimental section, we aim to evaluate the performance of our proposed model for relation extraction, focusing on its ability to integrate semantic and structural information of target entity pairs. The section is organized as follows: we first present the experimental setup, including the datasets, environment, and learning process. Next, we conduct experiments to analyze the influence of entity-related features and compare unit and fusion features. Furthermore, we compare our model with several

well-established methods to demonstrate its effectiveness in capturing the semantics and structural information of a target entity pair within sentences. Our findings provide valuable insights into the potential of our proposed model and its generalizability across different datasets.

## 5.1 Experimental setup

In this section, we present the experimental setup, including the datasets, environment, and learning process, to evaluate the performance of our proposed model for relation extraction.

### 5.1.1 Datasets

In this study, we evaluated the performance of our proposed model using three publicly available datasets: ACE05 English (WALKER et al. 2006), ACE05 Chinese (WALKER et al. 2006), and SanWen (Xu et al. 2017). The ACE05 datasets consist of diverse, non-domain-specific texts from radio, news broadcasts, and weblogs, and each instance is annotated with entity-related features and target relations. While the datasets provide valuable annotation features for extracting potential semantic associations and sentence structures, they exhibit a significant imbalance in the proportion of positive and negative cases, with a larger number of negative instances generated using predefined rules. The Chinese and English datasets contain 9,244 positive and 98,140 negative relation instances, and 6,583 positive and 97,534 negative relation instances, respectively. The SanWen dataset is a domain-specific Chinese relation extraction dataset with 13,462 training instances, 1,347 validation instances, and 1,675 test instances.

### 5.1.2 Environment

The experiments were conducted on a Linux-based system, utilizing TensorFlow version 1 as the deep learning framework and an A100-40 G graphics card.

To minimize noise influence and maximize relevant information, we fixed the length of the input sentence to 100 based on an analysis of the three datasets and a trend line graph of their length distribution (as illustrated in Figs. 4 and 5). Precision (P), recall (R), and F1-score (F1) are commonly used metrics in relation extraction. In addition, we also considered other metrics such as the area under the precision-recall curve to provide additional insights into the model's performance. Ultimately, P/R/F1 is employed to assess the model's performance, with P indicating the system's accuracy, R evaluating the system's recall, and F1 providing a comprehensive metric reflecting system

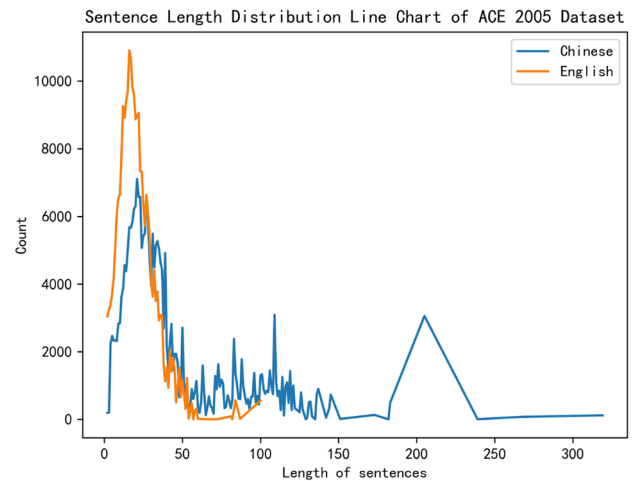


Fig. 4 Sentence lengths in ACE05

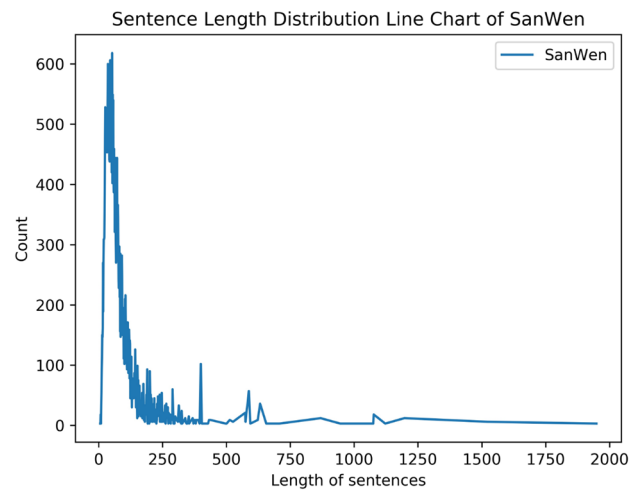


Fig. 5 Sentence lengths in SanWen

performance, derived from the harmonic mean of precision and recall. Explained as Eq. 10.

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (10)$$

### 5.1.3 Learning process

In this subsection, we describe the learning process of our proposed model for relation extraction.

Initially, for the representation of text, we employed two methods: 1) randomly initialized word vector lookup tables and 2) pre-trained word vector lookup files (Wiki-100 for Chinese, Google-News for English). We used different

settings in various experiments and provided corresponding instructions. We employed the AdamW optimization algorithm (Kingma and Ba 2014) with an initial learning rate of  $2e-5$  and a weight decay of  $1e-6$  to minimize the cross-entropy loss, and the cross-entropy loss function was used to calculate the loss during the model's learning process. The cross-entropy loss is defined as:

$$L(y, \hat{y}) = -\sum_{i=1}^C y_i \log \hat{y}_i \quad (11)$$

where  $y$  is the true label,  $\hat{y}$  is the predicted label, and  $C$  is the number of classes.

To prevent overfitting, we applied dropout (Srivastava et al. 2014) with a rate of 0.5 on the fully connected layers. We also used gradient clipping with a maximum gradient norm of 5 to avoid the exploding gradient problem. Our model was trained for a maximum of 100 epochs with early stopping if the validation loss did not improve for 20 consecutive epochs. We configured the batch-size parameter to 64, 32, and 32 for the training, validation, and testing processes, respectively.

During training, we monitored the loss and accuracy on the training and validation sets. The model with the best validation performance was selected and evaluated on the test set. This approach allowed us to observe the model's learning behavior and generalize its performance across different experimental settings.

In summary, this section provided a comprehensive description of our experimental setup, detailing the datasets used for evaluation, the computational environment, and the learning process implemented to train and assess our proposed model for relation extraction. The experimental results and discussion will be presented in the following section.

## 5.2 Experimental results and analysis

In this section, we present experiments conducted on three datasets to validate the effectiveness of our proposed model in integrating both semantic and structural information about the target entity pair for relation extraction. For word representation conversion, we employed the Wiki-100 word vector file for ACE05 Chinese and SanWen datasets and the Google-News word vector file for ACE05 English.

### 5.2.1 Influence of entity-related features

The purpose of this experiment is to investigate the impact of entity-related features on relation extraction using the ACE05 English dataset. To achieve this, we employed a convolutional neural network as the fundamental classifier model. Though the end-to-end methods have shown good performance on many tasks, such as relation extraction and other tasks of information extraction (Fei et al. 2020; Wang

**Table 3** Experimental results with entity-related features on ACE05 English

Method	Explanation	P (%)	R (%)	F1 (%)
1	Input raw sentence	75.14	52.80	62.02
2	Input sentence with entity marker	78.62	58.43	67.04
3	Input $U_1 - U_4$	84.22	62.43	71.71
4	Input $U_1 \sim U_5$	84.13	65.26	73.50
5	Input $U_1 - U_5$ with Google-News	86.01	67.62	75.71

and Lu 2020; Fei et al. 2021, 2022). In this experiment, in order to better explore and analyze the influences of performance brought by various entity-related features (unit features and fusion features in our study) in relation extraction while ruling out the improvement in model performance due to the superiority of the end-to-end approaches, we assessed five pipeline-style methods to examine the influence of the unit features on relation extraction. The results are displayed in Table 3. This can help us gradually introduce different entity-related features and show their contribution to relation extraction performance more clearly.

Method 1 solely used original sentences containing entity pairs as input, without incorporating auxiliary entity-related features to express structure and semantics of the target entity pair, resulting in suboptimal performance. Method 2 enhanced the performance by incorporating the entity marker  $U_5$  to capture entity structure. Method 3 integrated the semantics of the target entity by introducing unit features  $U_1 - U_4$ , improving performance compared to Method 2. Method 4 employed all unit features  $U_1$  to  $U_5$  to capture the target entity's semantic and structural information, resulting in superior performance. Finally, Method 5 introduced all unit features  $U_1$  to  $U_5$  and utilized a pre-trained Google-News word vector file to initialize the text representation, capturing abundant semantic and structural information and leading to the best performance.

The results suggest that incorporating diverse entity-related features enable the neural network to learn high-order abstract features of the sentence. As the unit features associated with the target entity pairs are progressively introduced from Method 1 to Method 5, the model captures both semantic and structural features about the target entity pair in the sentence, mitigating the feature sparsity in relation extraction. The experimental results demonstrate the efficacy of the features pertaining to the target entity pair for relation extraction.

### 5.2.2 Comparison of unit and fusion features

In this section, we aim to evaluate the effectiveness of unit features and fusion features within a unified network model. To ensure the validity and credibility of our



**Table 4** Experimental results under CNN on ACE05 English

Type	Unit features			Fusion features		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
PHYS	72.73	58.89	65.07	86.12	59.16	69.89
ART	94.60	54.70	69.30	93.12	62.11	76.12
<i>GEN – AFF</i>	72.35	50.15	58.95	62.18	52.25	56.60
<i>ORG – AFF</i>	93.86	77.30	84.66	95.62	80.32	87.28
<i>PART – WHOLE</i>	82.67	71.28	76.56	77.51	71.26	74.26
<i>PER – SOC</i>	88.67	77.60	82.80	98.43	81.50	89.65
<i>Macro – AVG</i>	84.13	65.26	73.50	86.22	67.98	76.21

**Table 5** Experimental results under CNN & GCN on ACE05 English

Type	Unit features			Fusion features		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
PHYS	82.18	50.92	62.88	81.72	63.09	71.18
ART	90.7	60.94	72.90	97.83	70.32	81.85
<i>GEN – AFF</i>	67.92	53.73	60.00	72.09	46.22	56.34
<i>ORG – AFF</i>	94.81	78.92	86.14	94.36	81.59	87.52
<i>PART – WHOLE</i>	76.74	75.86	76.3	79.35	79.35	79.35
<i>PER – SOC</i>	96.92	78.75	86.9	91.79	83.71	87.55
<i>Macro – AVG</i>	84.88	66.52	74.59	86.18	70.71	77.70

experiment, we use two specific models, namely, CNN and a combination of CNN and GCN. We validate our approach on the ACE05 English dataset and report our experimental results in Table 4 and 5. By comparing these results, we can determine which type of feature is more effective for relation extraction.

The experimental results are presented in Table 4 and were obtained using a CNN model on the ACE05 English dataset. When deploying the network model, we used a single-layer CNN with four convolutional kernels of size (2, 3, 4, 5) \* D (dimension of the embedding vector) to capture local contextual semantic dependencies. In the same model environment, fusion features exhibited superior performance to unit features, with an F1-score improvement of over 2.71%. Furthermore, even without using pre-trained word embedding files, the performance of fusion features still outperformed unit features that rely on representations obtained from pre-trained word embeddings by 0.5% F1-score, as demonstrated in Method 5 in Table 3 and the experimental results for fusion features in Table 4.

These experimental results demonstrate two phenomena. Firstly, fusion features that are composed of correlations between unit features can help the CNN model capture local sentence features and learn connections between entity features, thus learning semantic structural information in sentences. Secondly, in the context of pre-trained word embeddings enhancing the semantic information of text representations, this experiment found that fusion features can

even outperform enhanced text representations, providing deep insights for text representation-dependent tasks in the field of natural language processing.

In the experiments presented in Table 5, we used a deep learning model that combines CNN and GCN to learn the semantic structural information contained within both unit feature sequences and fusion feature sequences. When constructing the CNN structure, we still used a single-layer CNN with four convolutional kernels of size (2, 3, 4, 5) \* D (dimension of the embedding vector) to learn abstract representations from sentence feature sequences that embedded entity tags. For GCN, we also used a single-layer structure and constructed the matrix M based on the connection relationships between features. The weight matrix used to train the learning features was generated from randomly generated normal distribution values, which was used to learn abstract representations from entity-related feature sequences (excluding sentence sequences that embedded entity tags).

By analyzing the experimental data presented in Table 4 and 5, we can draw two interesting conclusions. Firstly, the performance of fusion features outperforms that of individual unit features in any model, indicating that fusion features can help deep learning models capture a considerable number of local features, including semantic structural information, and that embedding entity marker features can simultaneously enable the model to learn global features. These include semantic structural information that cannot be captured by unit features alone, demonstrating that the

**Table 6** Experimental results on ACE05 English

Model	Feature	F1(%)
Maximum Entropy (Kambhatla 2004)	Multiple features	52.80
Mix – CNN (Zheng et al. 2016)	Multi-grained local features	53.60
SVM (Zhou et al. 2005)	Phrase parsing	68.00
BERT (Wang and Lu 2020)	Table-sequence encoders	70.40
BERT (Zhong and Chen 2020)	Entity structure	73.10
Ours	Entity semantics & structure	77.70

correlation between entity-related features is more meaningful than their independence and can effectively improve relation extraction performance in a deep learning framework. Secondly, when comparing Table 4 and 5, we found that the F1-score of GCN was 1.09% higher than that of CNN when only using unit features. However, even when GCN extracted abstract features from the representation of unit feature sequence alone, the relation extraction performance was still 1.62% lower than that of a fusion feature in an independent CNN model, indicating that both fusion features and GCN have a positive impact on relation extraction. Therefore, combining GCN and fusion features achieves the best performance, with an F1-score of 77.70%. This experiment demonstrates the effectiveness of fusion features and GCN in neural network-based relation extraction.

Based on the experimental data presented in Table 3, Table 4, and Table 5, we can draw three conclusions. Firstly, entity-related features provide rich semantic information about entities and their semantic relationships to the model during identification. Secondly, fusion features capture the connections between unit features while providing the model with both textual semantic and structural information. Thirdly, GCN networks can integrate the internal relationships between entity-related feature sequences to fuse local and global information in sentences. In conclusion, utilizing GCN to combine the associated information between feature nodes plays a crucial role in identifying the semantic relationships between entities.

### 5.2.3 Model comparison

In this section, we present a comparative analysis of our proposed model with several well-established models in relation extraction, showcasing the effectiveness of our model in capturing the semantics and structural information of target entity pairs within sentences. We compare with models such as Kambhatla et al.'s feature-based maximum entropy (ME) model (Kambhatla 2004), Zheng et al.'s convolutional neural network (CNN) framework incorporating multiple convolutional kernels of varying sizes (Zheng et al. 2016), Zhou et al.'s support vector machine (SVM) model integrating multiple vocabularies, syntax, and semantic knowledge (Zhou et al. 2005), and Zhong et al.'s BERT pre-trained

**Table 7** Experimental results on ACE05 Chinese

Model	Feature	F1(%)
TRE (Alt et al. 2019)	Reusing of entity	56.20
Tree core (Yu et al. 2010)	Entity type, LDC, GPE	67.00
CNN (Nguyen and Grishman 2015)	Location of entity	67.30
CNN (Yang et al. 2021)	Reusing & tagging of entity	78.00
CNN + BERT (Yang et al. 2021)	Sentence	80.30
Att – BiLSTM (Zhou et al. 2016)	Solid structure	84.20
CNN + BERT (Yang et al. 2021)	Reusing & tagging of entity	89.80
Ours	Entity semantics & structure	90.12

language model with entity tags (Zhong and Chen 2020). Wang et al. Wang and Lu (2020) proposed table-sequence encoders where two different encoders—a table encoder and a sequence encoder—are designed to help each other in the representation learning process. The comparative experimental results for the ACE05 English dataset are shown in Table 6, further demonstrating the model's ability to capture the semantics and structure features while alleviating feature sparsity.

Earlier approaches in relation extraction involved constructing a syntactic–semantic relation tree based on the tree-kernel method for relation extraction, achieving a 67.0% F1-score on the ACE05 Chinese dataset (Yu et al. 2010). Nguyen TH et al. Nguyen and Grishman (2015) utilized a multi-kernel CNN to automatically extract text features, initializing it with a pre-trained word vector, incorporating position vectors to identify the target entity pair in the sentence, and achieving an F1-score of 67.3% on the ACE05 Chinese dataset.

However, CNN-based models may overlook long-distance dependencies in sentences. To address this issue, Zhou et al. (2016) proposed an attention mechanism-based bidirectional short-term memory network (Att-BiLSTM) to capture long-distance semantic dependencies in sentences. They introduced a method for marking target entity pairs in the text using entity markers such as “< e1>, </e1>”, achieving an F1-score of 84.10% on the ACE05 Chinese dataset.

**Table 8** Experimental results on SanWen

Model	Feature	F1(%)
SVM (Hendrickx et al. 2019)	Entity type, dependency, POS	48.90
RNN (Alt et al. 2019)	Entity type, POS	49.10
CNN (Zenget al. 2014)	Location feature, entity features	52.40
<i>CR – CNN</i> (Santos et al. 2015)	Location feature	54.10
<i>SDP – LSTM</i> (McDonough et al. 2019)	POS, entity features	55.30
DepCNN (Liu et al. 2015)	Word embedding, WordNet	55.20
BRCNN (Cai et al. 2016)	Word embedding + POS, NER, WordNet	55.60
<i>Att – BLSTM</i> (Zhang et al. 2020)	Character embedding + position embedding	53.30
<i>C – Att – LSTM</i> (Zhang et al. 2020)	Character embedding + position embedding, entity sense	56.20
MacBERT (Cui et al. 2020)	Character-based features	63.73
Ours	Entity semantics & structure	64.84

As shown in Tables 6 and 7, our proposed model outperforms the abovementioned models in terms of F1-score, effectively capturing the semantics and structural information of target entities in sentences by introducing fusion features. The model achieved a 4.6% improvement in F1-score compared to the comparative works, further substantiating the efficacy of our proposed model.

It is worth noting that previous approaches did not account for entity nesting features. By incorporating entity structure and nesting features, researchers utilizing randomly initialized word vectors achieved a 78.0% F1-score on the ACE05 Chinese dataset (Yang et al. 2021). Moreover, Yang et al. (2021) combined CNN and the BERT model to encode raw sentences in relation extraction, achieving an F1-score of 80.30%, proving the efficiency of strong language representations in relation extraction. Adding entity features and tagging to the BERT model further improved the performance by 9.5%, reaching an F1-score of 89.80%. Recently, Alt et al. (2019) adopted a transformer for relation extraction to learn implicit linguistic features solely from plain text corpora using unsupervised pre-training. Their model reached an F1-score of 56.20% on the ACE05 Chinese dataset.

Overall, the experimental results and model comparisons on ACE05 English and Chinese datasets demonstrate the effectiveness of our proposed model in capturing diverse abstract features and mitigating the issue of feature sparsity in relation extraction. The experimental results on the SanWen dataset further validate the generalizability of our proposed model.

In this section, we analyze the performance of our proposed model on the SanWen dataset and compare it with other popular relation extraction models and are presented in Table 8. The shallow-structure SVM combined with numerous external features was verified on this dataset (Hendrickx et al. 2019), and methods for automatically acquiring text features using neural networks were also evaluated (Alt et al. 2019; Zenget al. 2014; Santos et al.

2015; McDonough et al. 2019; Liu et al. 2015; Cai et al. 2016; Zhang et al. 2020). Employing models that extract information at multiple granularities for different features can effectively enhance the performance of relation extraction. For instance, Zhang et al. (2020) proposed a multi-feature fusion model that integrates multiple levels of features into a deep neural network model to capture word-level features and obtain more structural information. Facilitated by the large language model, Cui et al. (2020) proposed a whole word masking (WWM) strategy to improve upon Chinese RoBERTa, and Zhao et al. (2023) applied the model on SanWen dataset.

In this section, we analyze the performance of our proposed model on the SanWen dataset and compare it with other popular relation extraction models and are presented in Table 8. The shallow-structure SVM combined with numerous external features was verified on this dataset (Hendrickx et al. 2019), and methods for automatically acquiring text features using neural networks were also evaluated (Alt et al. 2019; Zenget al. 2014; Santos et al. 2015; McDonough et al. 2019; Liu et al. 2015; Cai et al. 2016; Zhang et al. 2020). Employing models that extract information at multiple granularities for different features can effectively enhance the performance of relation extraction. For instance, Zhang et al. (2020) proposed a multi-feature fusion model that integrates multiple levels of features into a deep neural network model to capture word-level features and obtain more structural information. Facilitated by the large language model, Cui et al. (2020) proposed a whole word masking (WWM) strategy to improve upon Chinese RoBERTa, and Zhao et al. (2023) applied the model on SanWen dataset.

Our proposed model extracts multi-granularity abstract features through different layers of the neural network to further improve performance. Specifically, our model captures both local and global features of sentences by combining convolution and graph convolution on the representations of the input sentence. The F1-score on the SanWen dataset

reached 64.84%. Although the proposed model does not achieve exceptionally high performance, it represents a crucial direction for future research to enhance the performance of domain-specific texts, given that the dataset pertains to the field of Chinese literature.

#### 5.2.4 Case study

To better illustrate the effectiveness of our proposed method in relation extraction, we provide a case study on a specific example, demonstrating how our model captures both semantic and structural information of the target entity pairs.

Consider the following sentence:

“The ferocity of COVID-19 has affected the physical health of many people”.

In this sentence, the target entity pair is (COVID-19, health), and the relation we want to extract is “Impact”. The model first recognizes the entity-related features, such as the positions of the entities and the words between them. The fusion features, which capture the connections between these unit features, are then computed to provide the model with a richer representation of the sentence’s semantics and structure.

The GCN component of the model is responsible for aggregating and integrating the global and local features of the input sentence. By combining the abstract features learned by the CNN and GCN components, our model is able to capture the long-distance dependencies in the sentence, such as the connection between “COVID-19” and “health” as well as the connection between “affected” and “health”.

In this specific case, our model successfully extracts the “Impact” relation between the target entity pair (COVID-19, health). The effectiveness of the model in this example can be attributed to its ability to capture both the semantic and structural information of the target entity pairs through the use of fusion features and GCN.

This case study demonstrates the potential of our proposed method in effectively extracting relations from sentences, taking into account both the semantics and structure of the target entity pairs. By incorporating fusion features and GCN, our model can tackle complex sentences with long-distance dependencies and provide valuable insights for relation extraction tasks in natural language processing.

## 6 Conclusion

The semantics and structure of the target entity pair are essential for relation extraction, but limited semantic elements lead to a feature sparsity problem. Our model extracts unit features of the target entity pair and combines them into a sequence of fusion features to obtain high-order abstract semantics and structure within a sentence using a deep learning framework (convolution and graph convolution neural networks).

The experimental results demonstrate the efficacy of our proposed model in incorporating multiple semantic features to extract structured knowledge in the sentence comprehensively. However, this study has some limitations. First, the model still relies on the external features introduced by the original dataset, indicating the potential for the neural network to automatically adapt features. Second, the model complexity is high, and its recognition accuracy and speed require improvements within simpler network models. Finally, the scope of the proposed method has not been fully verified, and its limitations in other NLP tasks remain unknown.

To address these limitations, our future research will focus on exploring breakthroughs in NLP tasks and constructing information extraction cloud platforms. By addressing these limitations and verifying the scope, we aim to provide a solid theoretical and technical foundation for building such cloud platforms. Therefore, the proposed method can potentially enhance the field of information extraction in various ways.

In conclusion, this study highlights the importance of considering the structure of the target entity pair and incorporating multiple semantic features to enhance relation extraction performance. Although the proposed model has limitations, it represents a crucial direction for future research in NLP tasks, as well as information extraction cloud platforms.

**Acknowledgements** Thanks to the editors and anonymous reviewers for their valuable suggestions and comments that made our final version of the paper perfect.

**Author contributions** YZ helped in writing—original draft, formal analysis, funding acquisition, project administration, and supervision; XY helped in data curation, investigation, methodology, and resources; YY helped in investigation, writing—review & editing; SD contributed to funding acquisition and supervision; JQ contributed to supervision, writing—review & data curation. All authors have read and agreed to the final version of the manuscript.

**Funding** This work is supported by the National Natural Science Foundation of China: 91746116; Science and Technology Project of Guizhou Province: [2018]5788.

**Availability of data and materials** Not applicable, as there are no relevant research data and materials in this article.

**Code availability** The code used in this study will be uploaded to a GitHub repository and is available to other researchers upon reasonable request.

## Declarations

**Conflict of interest** The authors declare no conflict of interest in this study.

**Consent to participate** All researchers participated voluntarily and were informed of all relevant aspects of the study.

**Consent for publication** The researchers consent to the use of the verses in this study.

**Ethical approval** Not applicable, as this study does not involve any ethical issues.

## References

- Alt C, Hübner M, Hennig L (2019) Improving relation extraction by pre-trained language representations. arXiv preprint [arXiv:1906.03088](https://arxiv.org/abs/1906.03088)
- Cai R, Zhang X, Wang H (2016) Bidirectional recurrent convolutional neural network for relation classification. In: proceedings of the 54th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp 756–765
- Church KW (2017) Word2vec. *Nat Lang Eng* 23(1):155–162
- Cui Y, Che W, Liu T, Qin B, Wang S, Hu G (2020) Revisiting pre-trained models for chinese natural language processing. arXiv preprint [arXiv:2004.13922](https://arxiv.org/abs/2004.13922)
- Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding
- Fei H, Ren Y, Ji D (2020) Boundaries and edges rethinking: an end-to-end neural model for overlapping entity relation extraction. *Inform Process Manage* 57(6):102311
- Fei H, Ren Y, Zhang Y, Ji D, Liang X (2021) Enriching contextualized language model from knowledge graph for biomedical information extraction. *Brief Bioinform* 22(3):110
- Fei H, Wu S, Li J, Li B, Li F, Qin L, Zhang M, Zhang M, Chua T-S (2022) Lasuie: unifying information extraction with latent adaptive structure-aware generative language model. *Adv Neural Inf Process Syst* 35:15460–15475
- Fei H, Ji D, Li B, Liu Y, Ren Y, Li F (2021) Rethinking boundaries: End-to-end recognition of discontinuous mentions with pointer networks. In: proceedings of the AAAI conference on artificial intelligence vol 35, pp 12785–12793
- Fei H, Wu S, Ren Y, Zhang M (2022) Matching structure for dual learning. In: international conference on machine learning pp 6373–6391. PMLR
- Fei H, Zhang M, Ji D (2020) Cross-lingual semantic role labeling with high-quality translated training corpus. arXiv preprint [arXiv:2004.06295](https://arxiv.org/abs/2004.06295)
- Fei H, Zhang Y, Ren Y, Ji D (2020) Latent emotion memory for multi-label emotion classification. In: proceedings of the AAAI conference on artificial intelligence vol 34, pp 7692–7699
- Hendrickx I, Kim SN, Kozareva Z, Nakov P, Séaghdha DO, Padó S, Pennacchiotti M, Romano L, Szpakowicz S (2019) Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. arXiv preprint [arXiv:1911.10422](https://arxiv.org/abs/1911.10422)
- Hu H, Sui H, Hu Q, Zhang Y, Hu Z, Ma N (2022) Runoff forecast model based on graph attention network and dual-stage attention mechanism. *J Comput Appl* 42(5):1607
- Kambhatla, N (2004) Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In: proceedings of the ACL interactive poster and demonstration sessions pp 178–181
- Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- Kumar S (2017) A survey of deep learning methods for relation extraction. arXiv preprint [arXiv:1705.03645](https://arxiv.org/abs/1705.03645)
- Li J, Fei H, Liu J, Wu S, Zhang M, Teng C, Ji D, Li F (2022) Unified named entity recognition as word-word relation classification. In: proceedings of the AAAI conference on artificial intelligence, vol 36, pp 10965–10973
- Liu Y, Wei F, Li S, Ji H, Zhou M, Wang H (2015) A dependency-based neural network for relation classification. arXiv preprint [arXiv:1507.04646](https://arxiv.org/abs/1507.04646)
- Liu C, Sun W, Chao W, Che W (2013) Convolution neural network for relation extraction. In: international conference on advanced data mining and applications pp 231–242. Springer
- Matthews J, Altman DG, Campbell M, Royston P (1990) Analysis of serial measurements in medical research. *BMJ* 300(6719):230–235
- McDonough K, Moncla L, Camp M (2019) Named entity recognition goes to old regime france: geographic text analysis for early modern french corpora. *Int J Geogr Inf Sci* 33(12):2498–2522
- Nguyen TH, Grishman R (2015) Relation extraction: Perspective from convolutional neural networks. In: proceedings of the 1st workshop on vector space modeling for natural language processing pp 39–48
- Niu F, Zhang C, Ré C, Shavlik JW (2012) Deepdive: web-scale knowledge-base construction using statistical learning and inference. *VLDS* 12:25–28
- Ojokoh B, Igbe T, Afolabi B, Daramola O (2023) A graph model with integrated pattern and query-based technique for extracting answer to questions in community question answering system. *Soc Netw Anal Min* 13(1):45
- Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Kenton L, Zettlemoyer L (2018) Deep contextualized word representations. In: Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: human language technologies 1 (Long Papers), New Orleans, Louisiana, pp 2227–2237
- Ranjan R, Vathsala H, Koolagudi SG (2022) Profile generation from web sources: an information extraction system. *Soc Netw Anal Min* 12:1–12
- Santos CND, Xiang B, Zhou B (2015) Classifying relations by ranking with convolutional neural networks. arXiv preprint [arXiv:1504.06580](https://arxiv.org/abs/1504.06580)
- Song R, Chen X, Hong Y, Zhang m, (2019) Combination of convolutional recurrent neural network for relation extraction. *J Chinese Inform Process* 33(10):64–72
- Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958
- Waheeb SA, Ahmed Khan N, Chen B, Shang X (2020) Machine learning based sentiment text classification for evaluating treatment quality of discharge summary. *Information* 11(5):281
- Walker C, Strassel S, J M (2006) Ace 2005 multilingual training corpus. progress of theoretical physics supplement 110( 110), 261–276
- Wang J, Lu W (2020) Two are better than one: Joint entity and relation extraction with table-sequence encoders. arXiv preprint [arXiv:2010.03851](https://arxiv.org/abs/2010.03851)



- Wang H, Qin K, Zakari RY, Lu G, Yin J (2022) Deep neural network-based relation extraction: an overview. *Neural Comput Appl* 1–21
- Wu S, Fei H, Ren Y, Ji D, Li J (2021) Learn from syntax: Improving pair-wise aspect and opinion terms extraction with rich syntactic knowledge. *arXiv preprint* [arXiv:2105.02520](https://arxiv.org/abs/2105.02520)
- Xu J, Wen J, Sun X, Su Q (2017) A discourse-level named entity recognition and relation extraction dataset for chinese literature text. *arXiv preprint* [arXiv:1711.07010](https://arxiv.org/abs/1711.07010)
- Yang W, Qin Y, Huang R (2021) Sentence structure acquisition method for chinese relation extraction. p 16
- Yu H, Qian L, Zhou G, Zhu Q (2010) Chinese semantic relation extraction based on unified syntactic and entity semantic tree. *J Chinese Inform Process* 24(5):17–23
- Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J (2014) Relation classification via convolutional deep neural network. In: *proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers* pp 2335–2344
- Zhang J, Hao K, Tang X-S, Cai X, Xiao Y, Wang T (2020) A multi-feature fusion model for chinese relation extraction with entity sense. *Knowl-Based Syst* 206:106348
- Zhang D, Wang D (2015) Relation classification via recurrent neural network. *arXiv preprint* [arXiv:1508.01006](https://arxiv.org/abs/1508.01006)
- Zhang C, Zhang X, Jiang W, Shen Q, Zhang S (2009) Rule-based extraction of spatial relations in natural language text. In: *2009 international conference on computational intelligence and software engineering* pp 1–4. IEEE
- Zhao Q, Gao T, Guo N (2023) A novel Chinese relation extraction method using polysemy rethinking mechanism. *Appl Intell* 53(7):7665–7676
- Zheng S, Xu J, Zhou P, Bao H, Qi Z, Xu B (2016) A neural network framework for relation extraction: learning entity semantic and relation pattern. *Knowl-Based Syst* 114:12–23
- Zhong Z, Chen D (2020) A frustratingly easy approach for entity and relation extraction. *arXiv preprint* [arXiv:2010.12812](https://arxiv.org/abs/2010.12812)
- Zhou G, Su J, Zhang J, Zhang M (2005) Exploring various knowledge in relation extraction. In: *proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)* pp 427–434
- Zhou P, Shi W, Tian J, Qi Z, Li B, Hao H, Xu B (2016) Attention-based bidirectional long short-term memory networks for relation classification. In: *proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short Papers)*, pp 207–212
- Zhu H, Lin Y, Liu Z, Fu J, Chua T-S, Sun M (2019) Graph neural networks with generated parameters for relation extraction. *arXiv preprint* [arXiv:1902.00756](https://arxiv.org/abs/1902.00756)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.