Joint Extraction of Organizations and Relations for Emergency Response Plans with Rich Semantic Information Based on Multi-head Attention Mechanism

Tong Liu
College of Computer Science and Engineering, Shandong University of Science and Technology, China
liu_tongtong@foxmail.com

Haoyu Liu
College of Computer Science and Engineering, Shandong University of Science and Technology, China
940024761@qq.com

Weijian Ni
College of Computer Science and Engineering, Shandong University of Science and Technology, China
niweijian@sdust.edu.cn

Mengxiao Si
College of Computer Science and Engineering, Shandong University of Science and Technology, China
33684091@qq.com

Abstract: *At present, deep learning-based joint entity-relation extraction models are gradually gaining the capability to accomplish complex tasks. However, research progress in specific fields is relatively slow. Compared with other areas, emergency plan text possesses unique characteristics such as high entity density, extensive text, and numerous professional terms. These features challenge some general models, which struggle to handle the semantic information of emergency plan text effectively. In response to this, the paper addresses the complex semantics of emergency plan text. It proposes a joint extraction model for emergency plan organization and relationship, based on the multi-head attention mechanism (MA-JE). This model enriches semantic information by obtaining contextual information from various perspectives and different levels. The aim is to deeply mine and utilize sentence semantic information through extensive feature extraction of emergency plan text. The proposed model and the baseline model were separately tested on the Chinese emergency response plan dataset. The results indicate that the proposed approach surpasses existing baseline models in the joint extraction of entities and their relations. Furthermore, ablation experiments were conducted to verify the effectiveness of each module within the model.*

Keywords: *Emergency plan, entity-relation joint extraction, deep learning, attention mechanism.*

Received October 18, 2022; accepted January 26, 2023
https://doi.org/10.34028/iajit/20/6/5

1. Introduction

Emergencies have always accompanied the development of human society, becoming more diverse and complex with the evolution of time and technological advancements. It's an undeniable fact that all corners of the world face the perpetual threat of natural disasters, public health incidents, accidental disasters, and other emergencies. Characterized by complex situations, sudden occurrences, serious harm, and wide-ranging impacts, emergencies can be as devastating as the outbreak of COVID-19 at the end of 2019 or the crash of China Eastern Airlines flight MU5735 in 2022. To respond to these emergencies in a timely and effective manner, governments at all levels in China have established corresponding Emergency Response Plans (ERPs). ERPs form the foundation and guarantee for responding to various emergencies. By formulating a corresponding management system, we can ensure that the relevant emergency response agencies can guide the development of emergency rescue operations in a quick, efficient, and orderly manner when a major accident occurs. This approach can minimize casualties, economic losses, and environmental damage resulting from the accident. Generally speaking, ERPs include general principles, an organizational command system, responsibilities, classifications, and many other elements. Among these, emergency agencies, serving as the main bodies executing the emergency plans, play a pivotal role [12]. Therefore, determining how to discern emergency organizations and their interrelationships within the vast array of ERPs is of immense practical significance. This understanding ensures the timely and accurate transmission of emergency information.

In early entity and relation extraction methods, the tasks of entity recognition and relation extraction [2] were conducted separately, a process known as the pipeline method [3]. This approach first identifies all entities present in a sentence, followed by pairing and matching these entities. However, three main issues commonly arise with this type of method [13]:

1. The two tasks are carried out independently of each

other, and the erroneous entities generated during the entity recognition process will be passed to the relation extraction module, resulting in the accumulation and propagation of errors.
2. When performing relation extraction on candidate entity pairs without semantic relations, these redundant entities will seriously affect the performance of relation extraction.
3. The lack of interaction between the two tasks ignores the inherent dependencies that exist between the two tasks.

Based on the analysis of the primary issues with the aforementioned pipeline method, this paper conducts research in the field of ERPs. The goal is to design a joint entity-relation extraction model suitable for ERP text, taking into account the characteristics of emergency response plan data. To tackle the problem of complex ERP text and numerous technical terms, the paper starts from the perspective of mining deep feature information. It proposes a joint extraction model for organizational structure and relationships based on the multi-head attention mechanism to enrich semantic information. By using a stacked double-layer convolution and multi-head attention mechanism in the embedding layer, an abundance of potential contextual semantics can be obtained from multiple angles and different levels. This approach addresses the issues of insufficient utilization of text information and single feature utilization. In the relation extraction module, a multi-segment convolution method is implemented to extract features from different parts of the text. This technique prevents the loss of semantic information that could occur due to missing parts of the text. The proposed model and the baseline model were separately tested on the Chinese ERPs dataset. The results indicate that the proposed method surpasses existing entity-relation joint extraction models. Furthermore, ablation experiments were conducted to verify the effectiveness and rationality of each module within the model.

2. Related Work

In 2016, deep learning techniques [6] were introduced to the field of entity-relation joint extraction [10, 16]. The joint modeling method based on deep learning eliminates the need for structural engineering design and can automatically learn sentence features that capture more complex internal structures. Thanks to its strong malleability, adaptability, portability, and other advantages, deep learning has gradually replaced feature-based methods. The two earliest models that applied deep learning to joint entity-relation extraction were the bidirectional LSTM (BiLSTM) dependency tree model [6] and the BiLSTM-CRF model [10], both proposed at the ACL conference in 2016. These models laid a solid foundation for the subsequent development of joint entity-relation extraction based on deep learning. Building upon this, Katiyar and Cardie [11] proposed a model combining the attention mechanism with BiLSTM, although it was only used to extract three types of entities and two types of relations related to "opinions". Based on the BiLSTM-CRF architecture, Nguyen and Verspoor [17] proposed a deep biaffine attention mechanism for dependency analysis, which was then used in the relationship classification module. The introduction of the attention mechanism helps to maximize the role of key information, thereby improving the extraction effect to a certain extent.

For the first time, Zheng *et al*. [27] transformed joint extraction into a labelling task to directly model triples. The encoding part utilized a BiLSTM structure, and the decoding part employed an LSTM & softmax structure. Subsequently, Zhou *et al*. [28] and Geng *et al*. [5] improved the model based on this three-part labelling scheme. Their sequence labelling-based methods significantly simplified the model complexity and reduced the impact of invalid entities on the model. However, these methods assign only one label to each word, and thus, they cannot address the problem of overlapping relationships.

To address the issue of entities having multiple relations, Bekoulis *et al*. [1] treated joint extraction as a multi-head selection problem, where the predicted relationship between the last word in each entity is considered as the predicted relationship between the entities. To enhance the robustness, Huang *et al*. [8] refined the model by replacing BiLSTM with BERT as the encoder. However, since the multi-head selection method relies solely on the last word of the entity for prediction in relation extraction, such methods often yield low accuracy [14]. Additionally, Zeng [24] introduced a joint extraction model with a replication mechanism, which is based on the encoder-decoder structure. Zeng *et al*. [23] further improved their own model and presented a multi-task learning framework with a replication mechanism. This effectively addressed the issues of the earlier models, but overfitting was observed in experiments conducted on their dataset. Both Eberts and Ulges [4] and Ji *et al*. [9] utilized a BERT pre-trained language model in the form of span-based negative sampling. They used local contextual information for joint extraction, which effectively alleviated the complexity of previous span-based models. Additionally, the method of joint extraction using hybrid networks has also gained popularity. For instance, combining CNN and RNN to capture the dependencies between labels [19, 20, 26], and introducing a Graph Convolutional Network (GCN) to enrich feature information [7, 15, 29]. Recently, several joint extraction models have performed well. These include the cascaded binary labeling framework proposed by Wei *et al*. [22], the Handshaking-kernel single-stage decoding proposed by Wang *et al*. [21], and the one-stage decoding proposed by Zhao *et al*. [25]. Although these models have achieved impressive experimental results, they all require the design of a complex labeling framework, which to a certain extent, increases the complexity.

3. Model

3.1. Task Description

Joint entity-relation extraction refers to the process of simultaneously identifying entities and the semantic relations existing between them from unstructured text. Named entities usually refer to proper nouns with specific meanings in the text. These entities can be a word or a phrase and the most commonly studied types include “person's name”, “place name” and “institution name”. Relationships refer to semantic associations between entities, such as part-whole relationships (Part-Whole), personal-social relationships (PER-SOC), and institutional affiliation (GPE-Affiliation, GPE-AFF). The extracted entity pairs and relations can form triples, which can be formally described as: Given an ERP $S=\langle w_1, w_2,\ldots,e_i,\ldots w_i,\ldots e_j,\ldots w_n\rangle$, the entity-relation triples can be formally described as $\langle e_i, r, e_j\rangle$, where $e_i,e_j \in S$ represent the head entities and tail entities of the triples, respectively; r belonging to the target relation set; $R+\{r1,r2,\ldots,rk\}$ represents the relations between entity pairs. The goal of this paper is to automatically identify all emergency organizations and their corresponding relations from ERPs texts.

3.2. Model Structure

With the emergence of deep learning in 2006, CNN, RNN, GCN and hybrid neural networks have been widely used in the joint extraction of entity relations and have achieved good results. Based on the combination of RNN and CNN, Geng *et al*. [5] introduced a multi-head attention mechanism to obtain rich semantic information. However, due to the particularity of the data in the field of ERPs, these general models cannot be fully applied to the text of emergency plans. Inspired by this, we start by with strengthening the connection between contexts, and extracts feature depth from multiple perspectives and different semantic subspaces. This paper proposes a joint extraction model of emergency plan organization and relationship based on the multi-head attention mechanism, the model framework is shown in Figure 1.

For domain-specific entity-relation extraction, it is crucial to fully utilize the semantic information of the sentence itself, and the model should be able to extract the text's semantic information without introducing external complex features. The model mainly includes four parts: the embedding layer, the encoding layer, the entity recognition layer, and the relation extraction layer. First of all, in the embedding layer, we design a lexical embedding module based on a double-layer CNN and a context embedding module based on a multi-head attention mechanism. The CNN is used to generate character vectors and word vectors to obtain different levels of input features, which can map input features to different context spaces to obtain multi-angle context features. The output of the lexical embedding module and the output of the context embedding module are concatenated to form a single feature.

In the entity recognition module, we use the LSTM and softmax decoding method to assign labels, mitigating the flaw that arises when BiLSTM alone ignores the correlation between labels. In the relation extraction module, the sentence is divided into multiple segments according to the location of the entity, and CNN is used for feature extraction from each segment. This approach helps to avoid losing important information that could be beneficial for relation classification. The rest of this chapter will introduce each module in detail and provide the specific algorithmic flow.

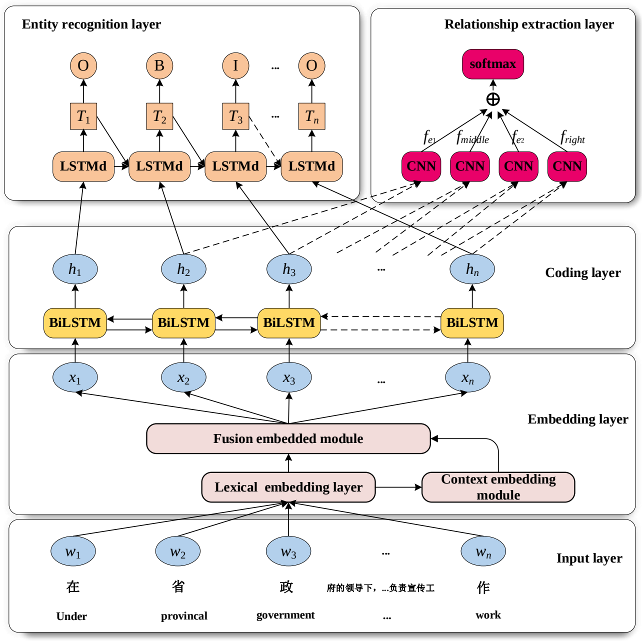


Figure 1. Framework of the MA-JE model.

3.2.1. Embedding Layer

Since the text of emergency plans is complex and contains many technical terms, understanding the sentences can be challenging. Therefore, feature extraction is particularly important. It can ensure that the key semantic information in the sentences is fully and correctly utilized, which is crucial for the subsequent relation extraction module. Drawing on the method of Geng *et al*. [5], the model in this paper introduces a lexical embedding module and a contextual embedding module to fully strengthen the connection between the contextual semantic information of the sentences. The embedding layer consists of three parts:

1) Lexical embedding module

Since Convolutional Neural Networks (CNNs) have a strong advantage in feature extraction, their central idea, which is to highlight features and extract key information in the text, is very popular in the field of natural language processing. Therefore, the lexical embedding module uses stacked double-layer CNNs for feature extraction. As the names of emergency plan organizations can be lengthy, the use of word vectors alone may ignore the dependencies between characters within a word. Additionally, relying solely on word vectors may not fully express the semantics of the entire word, which can lead to word isolation. Consequently, this module uses both character vector embedding and word vector embedding as input, effectively combining character and word information.

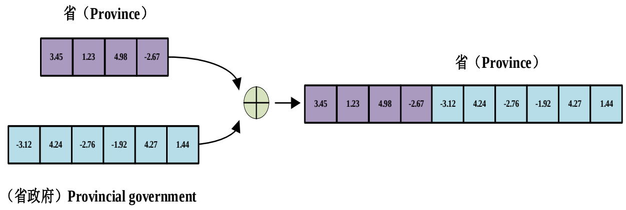


Figure 2. Example of character vector and word vector concatenating.

Given a sentence $S=[w_1, w_2, \dots, w_i, \dots, w_n]$, where n is the sentence length. The "Jieba" tool is used for tokenization, and then the Word2vec language model is used to train the input text. For the t-th word in the sentence, the character vector $x_t^{char} \in \mathrm{R}^{d^c}$ and the word vector $x_t^{word} \in \mathrm{R}^{d^w}$ are generated respectively, where d^c and d^w are the dimensions of the character vector and the word vector, respectively. Then the character vector and word vector are concatenated together as the final vector representation of each word. Figure 2 shows an example of the process of concatenating the character vector and the word vector. For instance, in the text "simultaneous/report/provincial government/", for the term "province" within the phrase "provincial government", its vector representation is composed of the character vector of the word "province" and the word vector of the phrase "provincial government".

Then, as shown in Figure 3, the resulting vector is passed through a stacked, two-layer convolutional neural network to obtain the final embedding representation x_t^c. By obtaining the vector representation of text from character-level features and word-level features, we can secure input features at different levels to enrich the semantic information of words.

2) Context embedding module

This module is designed with a multi-head attention mechanism. The term "multi-head attention" refers to the use of multiple sets of different linear transformations, which can map the input sequence to several different subspaces. This mechanism enables the model to explore the intrinsic interdependent features of sentence sequences from multiple perspectives, thus allowing the extracted features to obtain more levels of semantic information. The basic structural unit of the multi-head attention mechanism is the self-attention mechanism, which essentially perfects and supplements the self-attention mechanism. Multiple self-attention mechanisms are independent of each other in the calculation process. Taking the embedded representation of the previous module as input, it is mapped to different representation subspaces through different linear transformations, and concatenated with the weight matrices obtained from all subspaces, thus a new vector representation can be obtained. The calculation method is shown in Equation (1).

$$x_t^M = Multihead - Attention(Q^M, K^M, V^M) \tag{1}$$

Where Q^M is the query matrix and K^M is the key matrix and V^M is the value matrix.

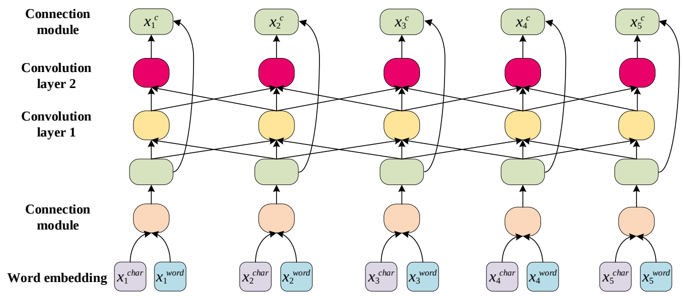


Figure 3. Model structure of the lexical embedding module.

3) Fusion embedding module

In order to obtain richer semantic information, the output vector of the lexical embedding module x_t^c is concatenated with the output vector of the context embedding module x_t^M, as shown in Equation (2), and finally a word vector representation with rich potential context information x_t is obtained.

$$x_t = [x_t^c; x_t^M] \tag{2}$$

This operation canconsider take into account both word-level features thereby, avoiding the loss of important features. Through the multi-head attention mechanism, the model can capture effective context semantic information of entities and relations can be captured from different semantic subspaces, preparing adequately for the subsequent decoding process.

3.2.2. Coding Layer

RNN [27] has strong applicability in sequence modeling, exhibiting high sensitivity to the order and position features of words. BiLSTM [10, 16], an optimization of RNN, has outstanding performance in relation extraction. Therefore, to better capture the bidirectional semantic dependencies of the input sentences, we use BiLSTM for encoding after the embedding layer. BiLSTM obtains the features of each word at the current moment by splicing the forward LSTM and the backward LSTM, thereby achieving the purpose of capturing past information and future information simultaneously.

In this layer, the output sequence of the embedding layer $[x_1, x_2, \ldots, x_n]$ is used as the input of BiLSTM. The forward LSTM captures sequence information from left to right, and the input at each moment x_t is combined with the hidden state of the previous moment $\vec{h}_{t-1}$ to obtain the forward hidden state vector $\vec{h}_t$. The calculation method is shown in Equation (3). The backward LSTM captures sequence information from right to left, combines the input at each moment x_t with the hidden state at the next moment $\overleftarrow{h}_{t+1}$, and obtains the backward hidden state vector $\overleftarrow{h}_t$. The calculation method is shown in Equation (4). The hidden vectors in the front and rear directions are concatenated to generate the final feature vector representation of each word h_t. The calculation method is shown in Equation (5). At this point, the hidden state output of the sentence sequence is represented as $[h_1, h_2, \ldots, h_n]$:

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}) \tag{3}$$

$$\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t+1}) \tag{4}$$

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \tag{5}$$

3.2.3. Entity Recognition Layer

In the entity recognition module, another variant of the LSTM structure-LSTMd, is used for decoding. Compared with the LSTM network structure, LSTMd also has three control gates, but the input gate in LSTMd has changed, and the rest of the units are the same as LSTM. The input gate of LSTMd not only receives the hidden state vector from the BiLSTM encoding layer h_t and the hidden state vector of the previous time step of LSTM s_{t-1}, but also adds the output label vector of the previous time step of LSTM T_{t-1}. The input gate formula of LSTMd is as follows:

$$i_t = \sigma(W_h h_t + W_s s_{t-1} + W_t T_{t-1} + b_i) \tag{6}$$

Among them, W_h, W_s, W_t, are the weight matrix and the bias vector b_i. T_t is the predicted label vector, s_t is linearly transformed by the following Equation:

$$T_t = W_{ts} s_t + b_{ts} \tag{7}$$

Where W_{ts} is the weight matrix and b_{ts} is the bias vector.

Finally, after the predicted label vector is subjected to a linear transformation, the softmax function is used to preform normalization calculation to obtain the probability distribution:

$$y_t = W_y T_t + b_y N_e S \tag{8}$$

$$p_t^i = \frac{\exp(y_t^i)}{\sum_{j=1}^{N_e} \exp(y_t^j)} \tag{9}$$

Where W_y is the weight matrix, b_y is the bias vector, and N_e is the number of entity labels.

The benefit of using the LSTMd structure in the entity recognition module is that LSTMd strengthens the connection between adjacent labels by treating the strong dependence of the previous word as an input to the current word, thus compensating for the limitation of using BiLSTM alone, which tends to ignore the correlation between labels during label prediction.

3.2.4. Relation Extraction Layer

For a given sentence, an entity set $[e_1, e_2, \ldots, e_m] \in \varepsilon$ is obtained through entity recognition, where m is the number of entities in the sentence. The next task involves combining the entity information, semantic information, and a given relation set R to extract the corresponding relation r for the entity pair, thereby forming a triplet $(e_1, r, e_2) \in \mathrm{T}$.

Firstly, according to the positions of the two organizations, the sentence is divided into five parts: the left context clause, entity one, the middle context clause, entity two, and the right context clause. The vectors of organization one and two are denoted as h_{e1} and h_{e2}. Given the characteristics of emergency plan data, the left context clause, which often lacks beneficial semantic information for relation extraction, is appropriately discarded. The remaining four parts undergo feature extraction via CNN, and the feature vectors are constructed through maximum pooling, which are recorded as f_{e1}, f_{middle}, f_{e2}, and f_{right}:

$$f_{e1} = CNN(h_{e1}) \tag{10}$$

$$f_{middle} = CNN[h_{e1}, h_{e1+1}, \ldots, h_{e2}] \tag{11}$$

$$f_{e2} = CNN(h_{e2}) \tag{12}$$

$$f_{right} = CNN[h_{e2}, h_{e2+1}, \ldots, h_n] \tag{13}$$

The obtained four feature vectors are concatenated into a vector $f = [f_{e1}; f_{middle}; f_{e2}; f_{right}]$, and then the softmax function is used to obtain the probability distribution of the relationship label:

$$y_r = W_f \cdot f + b_f \tag{14}$$

$$p_r^i = \frac{\exp(y_r^i)}{\sum_{j=1}^{N_r} \exp(y_r^j)} \tag{15}$$

Where W_f and b_f are the weight matrix and bias parameters, respectively, and Nr is the number of relation labels.

3.3. Pseudo Code

In summary, the MA-JE model can extract features from semantic subspaces of different levels and thoroughly utilize the crucial information from each part of the sentence through multi-segment CNN. In this section, we provide the pseudo code of the MA-JE model's process, as depicted in Algorithm (1).

Firstly, we concatenate the character vectors and the word vectors to form a vector representation, extract the features through a double-layer CNN and multi-head attention, and concatenate the final vector representation of each word. Then, we use LSTMd to decode and predict entity labels, and employ softmax to normalize the results into a probability distribution. The label with the largest probability is selected to form the entity set. After that, the sentence is divided into five parts according to the entity position of the organization, and the features are extracted by CNN to obtain their respective feature vectors. The final vector representation is obtained through concatenation. Finally, we use softmax to normalize the probability distribution and select the label with the highest probability to generate the triplet.

Algorithm 1: Process of the MA-JE model.

Input: *The text sequence S, the set of relations R*
Output: *Predicted triad* τ
Begin

1: *Initialize: S , R ,* τ
2: *for* $t \leftarrow 1$ *to* n *do*
3: *Concatenate character vectors and word vectors to form a vector representation:* $\left[x_t^{char}; x_t^{word}\right]$
4: *Extract feature by a two-layer CNN:* $x_t^c \leftarrow \left[x_t^{char}; x_t^{word}\right]$
5: *Use multi-headed attention to obtain feature depth r-e presentations from different subspaces:* $x_t^M \leftarrow x_t^c$
6: *Splice to get the final vector representation of each word* $x_t \leftarrow \left[x_t^c; x_t^M\right]$
7: *Obtain contextual representation by BiLSTM:* $h_t \leftarrow x_t$
8: *for* $t \leftarrow 1$ *to* n *do*
9: *Use LSRMd to predict entity labels:* T_t
10: *Use softmax to normalize to a probability distribu-ti on and choose the label with the largest probab-ility:* $p_t^i \sim softmax(y_t)$
11: *end for*
12: *Get the set of entities:* $\varepsilon \leftarrow \left[e_1, e_2, ..., e_m\right]$
13: *for* $e_1, e_2 \in \varepsilon$ *and* $e_1 \neq e_2$ *do*
14: *Divide the sentence into five parts, and extract features by CNN:* f_{e_1}、f_{middle}、f_{e_2}、f_{right}
15: *Concatenate the representation of each part to obt-a in the final vector representation:* $f \leftarrow \left[f_{e_1};\ f_{middle};\ f_{e_2};\ f_{right}\right]$
16: *Use softmax to normalize to a probability distribu-ti on and choose the label with the largest probab-ility:* $p_r^i \sim softmax(y_r)$
17: *end for*
18: $\tau \leftarrow (e_1, r, e_2)$
19: *end for*
20: *return predicted triad* τ
end

4. Experiment

4.1. Dataset

The datasets used in this experiment are sourced from the official websites of Chinese governments at all levels, ensuring the authenticity and reliability of the data sources. The data on these official websites is updated regularly, which ensures the timeliness of the data and the integrity of the emergency organization system [18]. Most of the information on these government websites is public. We use web crawling methods to collect emergency plan documents from these websites. Excluding the government websites of a few provinces that do not contain emergency plan data, we have crawled 3256 ERP files from 294 cities and saved them in TXT format. Considering that the organizational structure settings in the emergency plan of the same event type are similar, in order to identify as many organizational entities as possible in the emergency plan, we divide the 3256 emergency plan documents into 33 sub-categories. We then construct the emergency plan corpus through data preprocessing and data annotation processes. The dataset contains 16860 training documents and 4215 test documents, from 13 relation types, including the special relation “None”.

4.2. Evaluation Indicators

In this experiment, precision, recall and F1 value are used to evaluate the performance of the joint extraction model. The calculation formulas are shown in Equation (16), (17), and (18), respectively.

$$P = \frac{TP}{TP + FP} \tag{16}$$

$$R = \frac{TP}{TP + FN} \tag{17}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{18}$$

4.3. Model Comparison and Result Analysis

To test the performance of the proposed MA-JE model, it is compared with several baseline models, including SPTree model, NovelTagging model, BiLSTM+CNN model, and RS-Joint model.

1. SPTree [16]: a joint entity-relation extraction model based on the BiLSTM dependency tree structure. The

relation extraction layer completes the relation classification based on the shortest dependency path between paired entities.

2. NovelTagging [11]: this model designs a three-part tagging system that includes both entity information and relational information. It treats the entity-relationship joint extraction as a sequence labeling problem to directly model triples.
3. BiLSTM&CNN [26]: a joint entity-relation extraction method that uses a hybrid network of BiLSTM combined with CNN. In this model, CNN is used in the relational classification module to fuse entity pairs and their intermediate context information.
4. RS-Joint [5]: based on a three-part labeling strategy and integrating the entity-relationship joint extraction model of RNN and CNN, the multi-head attention mechanism and CNN are used to mine semantic information.
5. MA-JE: The model proposed in this paper.

Table 1. Experimental results of the comparison model.

Model	P	R	F1
SPTree	0.571	0.556	0.563
NovelTagging	0.647	0.424	0.506
BiLSTM&CNN	0.668	0.651	0.659
RS-Joint	0.761	0.563	0.647
MA-JE(ours)	0.817	0.785	0.801

Table 1 shows the experimental results of the proposed model in comparison with other baseline methods on the emergency plan text dataset. The following conclusions can be drawn:

1. The proposed model achieved 81.7% precision and 80.1% F1 value. However, the recall is relatively low.
2. The SPTree model uses the shortest dependency path, but because it relies on natural language processing tools, there is the phenomenon of error accumulation and propagation which greatly impacts its accuracy. The NovelTagging model uses tripartite tags to model triads, but its assumption that each entity participates in only one relationship limits its ability to extract overlapping triads well. This limitation is particularly problematic in the emergency plan text, which has numerous overlapping triads, leading to a low recall rate. Similarly, the RS-Joint model uses a three-part tag similar to NovelTagging, and although its embedding layer can enrich semantic information, its recall rate is still low. The BiLSTM&CNN model, which uses CNN in the relation extraction module to fuse entity pairs and their intermediate context information, overlooks the left and right sentences of the entity, resulting in the loss of some important information.
3. The comprehensive performance of the MA-JE model is superior to other baseline models, demonstrating that the embedding layer of the proposed model and the multi-segment CNN feature extraction play a crucial role. These elements enable the full utilization of sentence information, and prevent the loss of important information and incomplete semantics.

4.4. Ablation Experiment and Result Analysis

To evaluate the impact of each module of the proposed model on the final extraction results, certain components were either removed or replaced from the complete architecture. This allowed for an assessment of their effects on the joint extraction of entity-relationships. Specifically, two sets of neural network models were designed to conduct a series of ablation experiments.

In the first set of ablation experiments, the following three incomplete networks were tested to verify the effectiveness of each module in the embedding layer:

1. Lexical embedding module: Remove the lexical embedding module from the embedding layer, leaving only the context embedding module.
2. Contextual embedding module: Remove the contextual embedding module from the embedding layer, leaving only the lexical embedding module.
3. All: Remove both the lexical embedding module and the contextual embedding module from the embedding layer.
4. Multi-head: Replace the multi-head attention mechanism with a normal attention mechanism.

Table 2. Results of ablation experiments.

Model	P	R	F1
Lexical embedding module	0.776	0.724	0.749
Contextual embedding module	0.735	0.693	0.713
All	0.698	0.683	0.690
multi-head	0.805	0.762	0.783
Complete model	0.817	0.785	0.801

As Table 2 shows, the full network achieves better precision, recall, and F1 values. This demonstrates that each part of the embedding layer contributes to the model, with the specific conclusions as follows:

1. Compared with deleting the entire embedding layer, retaining only the lexical embedding module or the context embedding module improves the F1 value. However, the performance gap compared with the complete structure's F1 value is significant, proving that the embedding layer can enhance the model's performance by extracting features from different perspectives.
2. Retaining only the contextual embedding module yields better results than retaining only the lexical embedding module. This is because the multi-head attention mechanism within the contextual embedding layer can account for global information and discover multiple semantics of sentences.
3. When the multi-head attention mechanism of the context embedding module is replaced with a regular attention mechanism, better results are obtained

compared to other incomplete networks. This shows that the combination of the lexical embedding module and the attention mechanism is crucial for performance improvement.

In the second set of ablation experiments, the method proposed by the relationship extraction module in section 3.2.4, which segments sentences according to two organizational positions and uses CNN to extract features for each part, is examined. Four models are designed to study the effect of CNN on sentences, each performing the CNN operation on only the middle context clause and the right context clause:

1. Full-CNN: CNN is used to extract and concatenate the features of the left context clause, the 1st entity, the middle context clause, the 2nd entity, and the right context clause.
2. Middle-CNN: CNN is used to extract and concatenate only the features of the 1st entity, the middle context clause, and the 2nd entity.
3. Middle&Left-CNN: CNN is used to extract and concatenate only the features of the left context clause, the 1st entity, the middle context clause, and the 2nd entity.
4. Middle&Right-CNN: CNN is used to extract and concatenate only the features of the 1st entity, the middle context clause, the 2nd entity, and the right context clause.

Table 3 presents the experimental results from the four models described above. As the table illustrates, Middle&Right-CNN achieves higher precision, recall, and F1 values, whereas Middle&Left-CNN records the lowest scores in these categories. This confirms that in the realm of emergency plans, inter-entity context clauses and right context clauses are more critical for entity and relation identification, while left context clauses often do not contribute to semantic relationships. Additionally, when a method similar to that of Zheng *et al.* [27] is employed, which solely uses CNN for entity pairs and intermediate clauses, the experimental results do not align with their analysis. This demonstrates that the method proposed in this paper is more apt for the emergency plan dataset. When comparing Full-CNN with Middle&Right-CNN, using full sentences does not yield optimal results, further indicating that in contingency plan texts, most relations can be inferred from the entity pairs, the middle context clauses, and the right context clauses. Moreover, the inclusion of non-critical information can somewhat reduce identification accuracy.

Table 3. Comparison of experimental results of applying CNN to different parts of a sentence.

Model	P	R	F1
Full-CNN	0.765	0.741	0.753
Middle-CNN	0.732	0.749	0.740
Middle& Left-CNN	0.634	0.601	0.617
Middle& Right-CNN	0.817	0.785	0.801

5. Conclusions

This paper presents a joint extraction model of organizational structure and relationship based on a multi-head attention mechanism, aimed at enriching semantic information within the field of emergency response plans. An embedding layer is incorporated after the input layer, consisting of a lexical embedding module and a context embedding module. These modules capture and utilize rich contextual semantic information from various levels and perspectives. A series of experiments were conducted on the emergency plan dataset, and the results indicate that the proposed model outperforms several other baseline models. Additionally, two distinct sets of ablation experiments were designed to demonstrate the effectiveness of each module.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 71704096, Shandong Provincial Natural Science Foundation under Grant ZR2022MF319, the Qingdao Philosophy and Social Sciences Planning Project under Grant QDSKL1801122 and Grant QDSKL2001117, and the Talented Young Teachers Training Program of Shandong University of Science and Technology under Grant BJ20211110.

References

[1] Bekoulis G., Deleu J., Demeester T., and Develder C., “Joint Entity Recognition and Relation Extraction as a Multi-Head Selection Problem,” *Expert Systems with Applications*, vol. 114, pp. 34-45, 2018. https://doi.org/10.48550/arXiv.1804.07847

[2] Cowie J. and Lehnert W., “Information Extraction,” *Communications of the ACM*, vol. 39, no. 1, pp. 80-91, 1996. https://doi.org/10.1145/234173.234209

[3] E Hai-Hong., Zhang W., and Xiao S., “Research Review of Entity Relation Extraction Based on Deep Learning,” *Journal of Software*, vol.30, no.6, pp. 26, 2019.

[4] Eberts M. and Ulges A., “Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training,” *in Proceedings of the 24th European Conference on Artificial Intelligence*, Santiago de Compostela, pp. 2006-2013, 2020. https://doi.org/10.48550/arXiv.1909.07755

[5] Geng Z., Zhang Y., and Han Y., “Joint Entity and Relation Extraction Model Based on Rich Semantics,” *Neurocomputing*, vol. 429, pp. 132-140, 2021. https://doi.org/10.1016/j.neucom.2020.12.037

[6] Hinton G. and Ruslan R., “Reducing the Dimensionality of Data with Neural Networks,”

Science, vol. 313, no. 5786, pp. 504-507, 2006. DOI: 10.1126/science.1127647

[7] Hong Y., Liu Y., Yang S., Zhang K., and Hu J., "Joint Extraction of Entities and Relations Using Graph Convolution over Pruned Dependency Trees," *Neurocomputing*, vol. 411, pp. 302-312, 2020. https://doi.org/10.1016/j.neucom.2020.06.061

[8] Huang W., Cheng X., Wang T. and Chu W., "BERT-Based Multi-Head Selection for Joint Entity-Relation Extraction," *in Proceedings of the Natural Language Processing and Chinese Computing*, Dunhuang, pp. 713-723, 2019. https://doi.org/10.48550/arXiv.1908.05908

[9] Ji B., Yu J., Li S., Ma J., and Wu Q., "Span-Based Joint Entity and Relation Extraction with Attention-Based Span-Specific and Contextual Semantic Representations," *in Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, pp. 88-99, 2020. 10.18653/v1/2020.coling-main.8

[10] Katiyar A. and Cardie C., "Investigating Lstms for Joint Extraction of Opinion Entities and Relations," *in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, pp. 919-929, 2016. DOI: 10.18653/v1/P16-1087

[11] Katiyar A. and Cardie C., "Going Out on A Limb: Joint Extraction of Entity Mentions and Relations without Dependency Trees," *in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, pp. 917-928, 2017. DOI: 10.18653/v1/P17-1085

[12] Liu T., "Studies on Scenes' Construction of Emergency Planning-Part Ⅰ of Emergency Planning Technology Based on Scene-Task-Ability," *China Safety Production Science and Technology*, vol. 8, no. 4, pp. 5-12, 2012.

[13] Liu Y., "Joint Extraction of Entities and Relations Based on Multi-Label Classification," *in Proceedings of the IEEE 4th International Conference on Data Science in Cyberspace IEEE*, Hangzhou, pp. 106-111, 2019. DOI: 10.1109/DSC.2019.00024.

[14] Lin Y., Shen S., Liu Z., and Luan H., "Neural Relation Extraction with Selective Attention over Instances," *in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, pp. 2124-2133, 2016. DOI:10.18653/v1/P16-1200

[15] Miao L., Zhang Y., Xie B., and Li Y., "Joint Entity Relation Extraction Based on Graph Neural Network," *Computer Application Research*, vol. 39, no. 02, pp. 424-431, 2022.

[16] Miwa M. and Bansal M., "End-to-End Relation Extraction Using LSTMS on Sequences and Tree Structures," *in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, pp. 1105-1116, 2016. https://doi.org/10.48550/arXiv.1601.00770

[17] Nguyen D. and Verspoor K., "End-to-End Neural Relation Extraction Using Deep Biaffine Attention," *in Proceedings of the 41st European Conference on Information Retrieval*, Cologne, pp. 729-738, 2019. https://doi.org/10.48550/arXiv.1812.11275

[18] Ni W., Shen Q., Liu T., Zeng Q., and Xu L., "Generating Textual Emergency Plans for Unconventional Emergencies-a Natural Language Processing Approach," *Safety Science*, vol. 160, pp. 106047-106065, 2023. https://doi.org/10.1016/j.ssci.2022.106047

[19] Sun C., Wu Y., Al-Ajarmah M. and Sun S., "Extracting Entities and Relations with Joint Minimum Risk Training," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Brussels, pp. 2256-2265, 2018. DOI: 10.18653/v1/D18-1249

[20] Tan Z., Zhao X., Wang W. and Xiao W., "Jointly Extracting Multiple Triplets with Multilayer Translation Constraints," *Proceedings of AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 7080-7087, 2019. https://doi.org/10.1609/aaai.v33i01.33017080

[21] Wang Y., Yu B., Zhang Y., Liu T., and Zhu H., "Tplinker: Single-Stage Joint Extraction of Entities and Relations Throughtoken Pair Linking," *in Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, pp. 1572-1582, 2020. https://doi.org/10.48550/arXiv.2010.13415

[22] Wei Z., Su J., Wang Y., Tian Y., and Cheng Y., "A Novel Cascade Binary Tagging Framework for Relational Triple Extraction," *in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1476-1488, 2020. 10.18653/v1/2020.acl-main.136

[23] Zeng D., Zhang R., and Liu Q., "Copy Mtl: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning," *Proceedings of AAAI Conference on Artificial Intelligence*, vol. 34, no. 5, pp. 9507-9514, 2020. https://doi.org/10.48550/arXiv.1911.10438

[24] Zeng X., Zeng D., He S., Kang L., and Zhao J., "Extracting Relational Facts by an End to End Neural Model with Copy Mechanism," *in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, pp. 506-514, 2018. DOI: 10.18653/v1/P18-1047

[25] Zhao K., Xu H., Cheng Y., Li X., and Gao K., "Representation Iterative Fusion Based on Heterogeneous Graph Neural Network for Joint Entity and Relation Extraction," *Knowledge-Based Systems*, vol. 219, pp. 106888-106897,

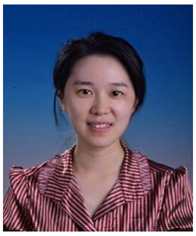
2021. https://doi.org/10.1016/j.knosys.2021.106888

[26] Zheng S., Hao Y., Lu D., Bao H., and Xu J., "Joint Entity and Relation Extraction Based on a Hybrid Neural Network," *Neurocomputing*, vol. 257, pp. 59-66, 2017. https://doi.org/10.1016/j.neucom.2016.12.075

[27] Zheng S., Wang F., Bao H., Hao Y., and Zhou P., "Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme," *in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, pp. 1227-1236, 2017. 10.18653/v1/P17-1113

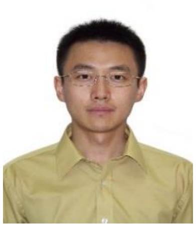
[28] Zhou Y., Huang L., Guo T., Hu S., and Han J., "An Attention-Based Model for Joint Extraction of Entities and Relations with Implicit Entity Features," *in Proceedings of the World Wide Web Conference*, San Francisco, pp. 729-737, 2019. https://doi.org/10.1145/3308560.3317704

[29] Zohaib A., Muhammad Q., Aasma M., and Parveen M., "Echo State Network Optimization Using Hybrid-Structure Based Gravitational Search Algorithm with Square Quadratic Programming for Time Series Prediction," *The International Arab Journal of Information Technology*, vol. 19, no. 3A, pp. 530-535, 2022. 10.34028/iajit/19/3A/13



Tong Liu is an associate professor and master tutor of the College of Computer Science and Engineering of Shandong University of Science and Technology. Her research interests include machine learning, data mining, knowledge mapping and artificial intelligence.

Haoyu Liu is a postgraduate student in the College of Computer Science and Engineering of Shandong University of Science and Technology, who has won many postgraduate scholarships, and his main research direction is natural language processing.



Weijian Ni received his doctorate in computer application from Nankai University in 2008. He is currently a professor at the College of Computer Science and Engineering, Shandong University of Science and Technology. His research interests include natural language processing, personalized recommendation, in-depth learning, business process mining and software engineering.

(Corresponding author: niweijian@sdust.edu.cn)

Mengxiao Si received a master's degree from the College of Computer Science and Engineering of Shandong University of Science and Technology in 2022. Her main research direction is natural language processing