Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **2**, May **2025**

Attention Graph Neural Network-based Relational Extraction of Legal Texts

Li Su

College of Big Data and Artificial Intelligence
Anhui Xinhua College, Hefei 230000, P. R. China
82415872@qq.com

Jiao Song*

College of Big Data and Artificial Intelligence
Anhui Xinhua College, Hefei 230000, P. R. China
499405428@qq.com

Yun-Fei Zhu

Northwestern University, Chiang Mai 20260, Thailand
1468059565@qq.com

*Corresponding author: Jiao Song
Received May 9, 2024, revised September 27, 2024, accepted December 12, 2024.

Abstract. *Intending to the issue that the existing legal text relationship extraction model can only obtain part of the text features, this article designs a legal text relationship extraction model relied on attention graph neural network to improve the comprehensiveness and accuracy of the legal text relationship extraction, and to realize the goal of high-performance relationship extraction for the domain dataset. Firstly, the legal text feature embedding representation is performed in the data preprocessing part, and the updating of the case description text graph nodes is realized through the message passing mechanism. Then the constructed text graph is encoded adopting the BERT pre-training model, and the text features are extracted using different feature extraction modules for the graph channel and the sequence channel respectively, the graph channel uses Graphical Convolutional Neural (GCN) network for feature extraction of entity nodes, and the sequence channel uses Convolutional Neural Network (CNN) for feature extraction of sentence and word nodes, and secondly, the features of the two channels are fused and enhanced using the attention mechanism. Finally, at the classifier layer, the final classification output of legal text entity relations is performed. The experimental outcome indicates that the Accuracy, Macro-F1 and Micro-F1 of the proposed model in this paper are 0.937, 0.694 and 0.884, respectively, which are better than the comparison model, proving the efficiency of the suggested model.*
Keywords: Attention mechanism; Graph neural network; Relational extraction; BERT model; Convolutional neural network

1. **Introduction.** As the artificial intelligence technology continuously growing, the automated processing and analysis of legal texts can improve the work efficiency of judicial administrators, and the integration of machine learning into the work of the legal field has been a research hotspot [1, 2]. The legal field involves a huge number of legal provisions, case precedents, regulations and other textual information, and there are complex relationships between these texts, such as the cross-references between legal provisions,

the correlation between cases and regulations, and so on. However, due to the complexity and rigor of legal texts, traditional manual reading and analysis methods are very time-consuming and laborious, so automated relational extraction techniques need to be developed to enhance analysis efficiency and accuracy [3, 4]. Currently, relational extraction techniques for natural language processing are able to quickly extract key information from a large variety of legal text data, but there are still many challenges in complex legal texts, such as relational extraction [5, 6, 7]. Legal text relationship extraction refers to the process of automatically identifying and extracting entities (such as names of people, places, organizations, etc.) and their relationships (such as employment relations, ownership relations, etc.) from legal documents. Thus, it is of great research significance to find out how to utilize the huge amount of legal texts more effectively and extract useful information from them accurately.

1.1. **Related work.** Hachey and Grover [8] first proposed the use of syntactic analysis tools, which can analyze the composition of legal text statements by following rules, but suffers from the problem of fragmentation and inaccuracy. Chen [9] proposed the use of an effective remotely supervised neural network for the extraction of legal text relationships and noise reduction of extracted data. Naseri et al. [10] proposed a sequential co-clustering algorithm to co-cluster extracted legal text templates and entity pairs, but it requires a lot of manpower and resources to refine the templates. Thomas and Sangeetha [11] combined syntactic analysis and pattern matching to extract relationships between legal text statements. Tran [12] encoded the legal text into a low-finesse semantic space to extract relationships from the text vectors. vector to extract relationships. With the development of machine learning methods, researchers have started to improve the relationship extraction methods. Ren et al. [13] used a maximum entropy-based machine learning method to extract relationships between statements by combining syntactic, semantic, and lexical analysis of legal texts. Förster et al. [14] proposed a method integrating a relationship model and a relationship tree to extract domain-entity relationships in legal texts, which realized the relationship between criminal networks and their relationships. The rise of deep learning, to a certain extent, alleviates researchers' reliance on natural language extraction tools, and gradually began to apply neural networks to legal text relationship extraction tasks. The advantage of neural network in the task of extracting legal text relations lies in its powerful feature learning ability, which can automatically learn and extract useful patterns and relations from a large number of legal text data without manually designing complex feature extraction rules. Chen et al. [15] designed a method for legal text relationship extraction using LSTM, taking word vectors as the input, and Convolutional Neural Network (CNN) will be used to learn the laws and ways of semantic combinations of sentences. In order to obtain semantic features that can better represent the relationship categories, Chen et al. [16] offered extracting features from legal text using deep CNNs, retaining as much information as possible in the sentences, and utilizing decision trees for parameter training of the model so as to predict the labels of the unknown packets. Gambhir and Gupta [17] added a single-head attentional mechanism, which further improves the CNN models relationship extraction. Vuong et al. [18] suggested a deep neural network model framework for legal text relationship extraction, but only some of the text features can be obtained. Chen et al. [19] designed a BERT-based legal text relationship extraction model with feature extraction based on convolutional neural networks. In order to improve the efficiency of relationship extraction, Bi et al. [20] offered a legal text relationship extraction model based on graph spectral theory and Graph Convolutional Neural (GCN) network, which aggregates the features of k-th order neighbors on the graph for computation, but the amount of computation is large. Feng et al. [21] used the graph attention mechanism to capture the effect of the main text context of the law on the node updating, but the features of the key content are fuzzy.

1.2. **Contribution.** Overall, there is still a huge room for progress in the research of extraction modeling of legal text relationships. Aiming at the issue of insufficient extraction of legal text features in the current research, this article designs a legal text relationship extraction model relied on attention graph neural network. First of all, the data preprocessing part transforms the text into a text graph, so that each legal text corresponds to a separate text graph, and adopts the message passing mechanism to converge the information of neighboring nodes and complete the update of its own node. It also adopts the gating mechanism to combine the information of the node itself with the information of the neighboring nodes, and finally realizes the node update. Then the BERT pre-training model is used to encode the constructed text graph to better obtain the semantic information of the text context. Secondly, the legal text features are extracted using dual channels, the graph channel uses GCN for entity nodes, and the sequence channel uses CNN for sentence and word nodes to extract features and mine text association information. Subsequently, the features of the two channels are augmented using the self-attention mechanism and gated attention mechanism respectively. Finally, at the classifier layer, the final classification output of legal text entity relations is performed.

2. **Basic theory.**

2.1. **Attention mechanism.** The essence of the attention mechanism is a process of addressing through a map-like structure. Given a query vector q related to a core task, the value of the attention mechanism is obtained by calculating the attention weights corresponding to the required keys [22]. Compared with inputting all the information into the neural network, this process can reduce the overhead of the neural network model by only obtaining the required keys for input [23], and its model structure is implied in Figure 1. The process of the attention mechanism is mainly divided into three steps as follows.

(1) Information input. Suppose a sequence of vectors $X = [x_1, x_2, ..., x_m]$ is input to an attention mechanism.

(2) Calculate the attention weight corresponding to *key*. Assuming that $key = value = X$, then getting the equation of attention distribution as follows.

$$\beta_i = \text{softmax}\big(s(key_i, p)\big) = \text{softmax}\big(s(X_i, p)\big) \tag{1}$$

where β_i is called the attention distribution and $s(X_i, p)$ is called the attention evaluation mechanism, which is generally obtained by using the point product method. Then the obtained attention weights are normalized by softmax function to make $\sum_{i=1}^{m} \beta_i = 1$, and $\beta_i \in [0, 1]$. The normalized weights are then used as the weights of the attention distributions.

(3) Information weighted average. A query through the context, the degree of attention received by the current vector, the higher the degree of attention the higher the value, and then use the soft attention [24] way to encode the input vector sequence, and finally get the result based on the attention mechanism after the coding of the vector, as follows.

$$S = att(p, X) = \sum_{i=1}^{M} \beta_i X_i \tag{2}$$

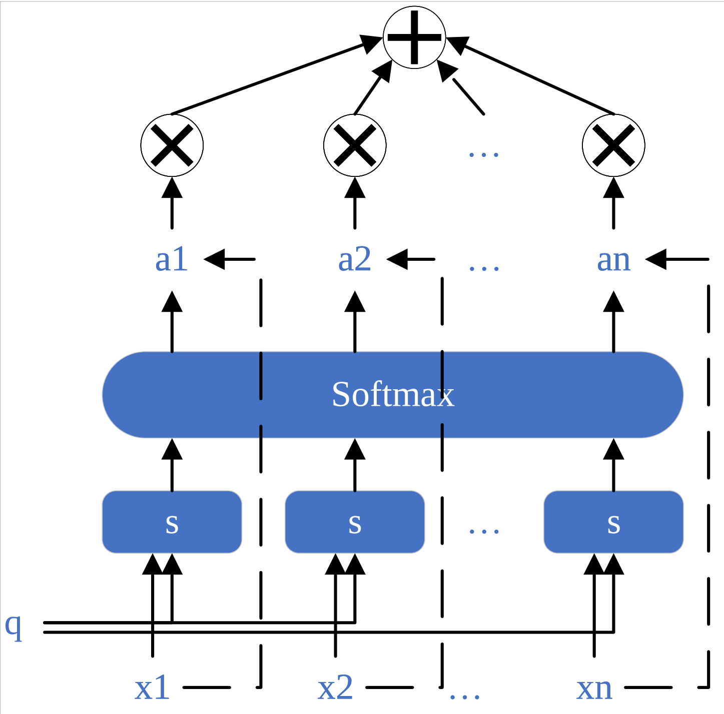

Figure 1. Model structure of attentional mechanism

2.2. **Graph neural network.** Deep learning methods based on CNN, RNN, etc. have achieved remarkable results in text feature extraction in the field of natural language processing, however, text features not only contain the sequence structure of words but also include a large amount of non-structured information, such as grammatical relations, referential relations, etc., which are hard to be expressed directly and are represented by graphical structural information [25].

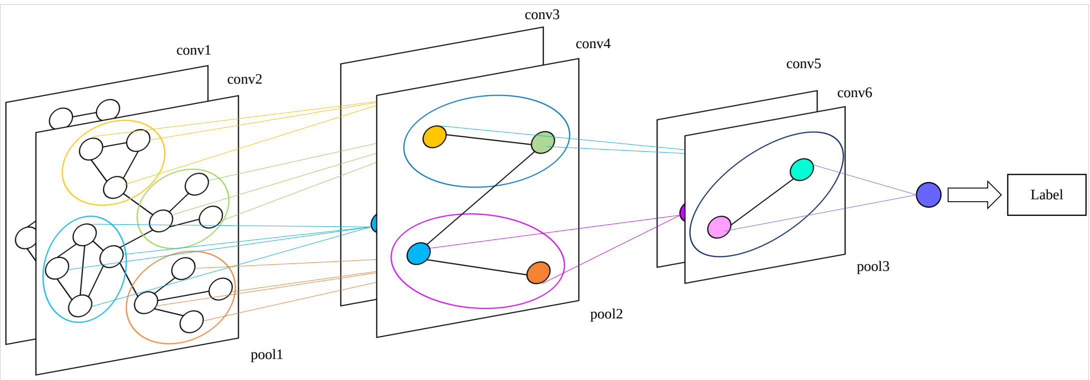

Figure 2. The process of realizing classification of text graph by graph neural network

The GCN network model [26], as an important branch of graph neural networks, performs convolution operations on graph structure data according to the Laplace matrix to obtain the extraction of features of graph structure data. The process of extracting features and realizing classification of text graph by graph neural network is implied in Figure 2. The calculation equation for the convolution operation on the text map is indicated in Equation (3) and Equation (4).

$$\hat{g}_i^k = \sum_{j=1}^{m} B_{ij} V_k h_j^{k-1} \tag{3}$$

$$g_i^k = ReLU\left(\frac{\hat{g}_i^k}{c_i + 1} + a_k\right) \tag{4}$$

where $\hat{g}_i^k$ is the temporary state representation of the i-th node of the k-layer GCN, h_j^{k-1} is the temporary representation of the j-th node of the $k-1$-layer GCN, g_i^k is the final representation of the i-th node of the k-th layer GCN, $c_i = \sum_{j=1}^{m} b_{ij}$ denotes the degree of the i-th node, i.e., the number of nodes that are directly connected to the i-th node, and V_k and a_k are the weight matrix and bias term of the k-th layer GCN, respectively.

3. **Representation of legal texts based on graph structures.** Before extracting legal text relationships, the text needs to be transformed into a text graph in the data preprocessing section, so that each legal text corresponds to a separate text graph. There are two kinds of nodes in the text graph: document nodes and word nodes, and the edges between the word nodes are obtained by the sliding window counting the word co-occurrence information in each text. The edges between the document nodes and the word nodes have the weight of 1, and the edges between the document nodes and the word nodes are established by the sliding window to gain the global information.

This article adopts the message passing mechanism [27] to aggregate the information of the neighboring nodes and complete the update of own nodes. The normalization operation is performed to achieve that the implied state information obtained which is the weighted average of the information of the neighboring nodes, and the calculation process is implied as bellow.

$$A^{s+1} = MLP^{s+1}\left(C^{-1} B G^s\right) \tag{5}$$

where $G^s \in \mathbb{R}^{m \times c}$ is the node feature representation matrix of the legal text graph in the graph (m is the number of nodes and c is the dimension of the initialized word vectors), $B \in \mathbb{R}^{m \times m}$ is the adjacency matrix of the text graph, $C \in \mathbb{R}^{m \times m}$ is the degree matrix of the text graph, and A^{s+1} is the weighted average of the neighboring nodes that were obtained.

After calculating the weighted average of the neighboring node information, the gating mechanism [28] is used to complete the combination of the node's own information with the neighboring node information, and finally the node update is achieved.

$$T^{s+1} = \delta\left(V_T^{s+1} A^{s+1} + W_T^{s+1} G^s\right) \tag{6}$$

$$Z^{s+1} = \delta\left(V_Z^{s+1} A^{s+1} + W_Z^{s+1} G^s\right) \tag{7}$$

$$\tilde{G}^{s+1} = \tanh\left(V_Z^{s+1} A^{t+1} + W^{s+1}\left(T^{s+1} \odot G^s\right)\right) \tag{8}$$

$$G^{s+1} = \left(1 - Z^{s+1}\right) \odot G^s + Z^{s+1} \odot \tilde{G}^{s+1} \tag{9}$$

where V and W are trainable weights, T is the reset gate and Z is the update gate. After S message passes, the final text graph node feature representation matrices $\tilde{G}^{s+1}$ and G^s are obtained. $\tilde{G}^s \in \mathbb{R}^{(m-1) \times c}$ is the node feature representation matrix that contains only word nodes and $G^s \in \mathbb{R}^{m \times c}$ contains text nodes.

To embed text, it is necessary to transform the text graph into text embedding representation vector. In the readout layer, the node representation matrix A of the text graph is computed using the attention mechanism to obtain the intermediate text vector B, and the intermediate text vector is spliced with the text vector representation in layer S to obtain C. The specific computations are indicated in Equation (10) and Equation (11).

$$\beta_i^S = \frac{\exp\left(\tanh\left(\hat{G}_i^S V_V^S\right) \cdot u^S\right)}{\sum_{i=1}^{n-1} \exp\left(\tanh\left(\hat{G}_j^S V_V^S\right) \cdot u^S\right)} \tag{10}$$

$$u^S = \left(\sum_{i=1}^{n-1} \beta_i^S\right) \cdot \hat{G}^S \tag{11}$$

where $\hat{G}^S$ is the representation matrix of the document nodes and $u^S \in \mathbb{R}^c$ is the trainable vector.

In the process of each message delivery, the node feature information of the text graph is gradually enriched, but in the process of enriching the information may carry part of the noise and these noises will affect the next delivery, so the text representation vectors generated in each message delivery are crucial, so the intermediate text vectors generated in each layer are spliced to get the final legal text representation $g^S \in \mathbb{R}^{S\times 2c}$, as indicated in Equation (12).

$$D^S = concat\big(readout(G^S) \mid s = 1, \ldots, S\big) \tag{12}$$

4. Legal text relationship extraction based on attention graph neural network.

4.1. Legal text encoding based on the BERT Model. To address the issue of insufficient feature extraction in existing legal text extraction methods, the BERT pre-training model is first

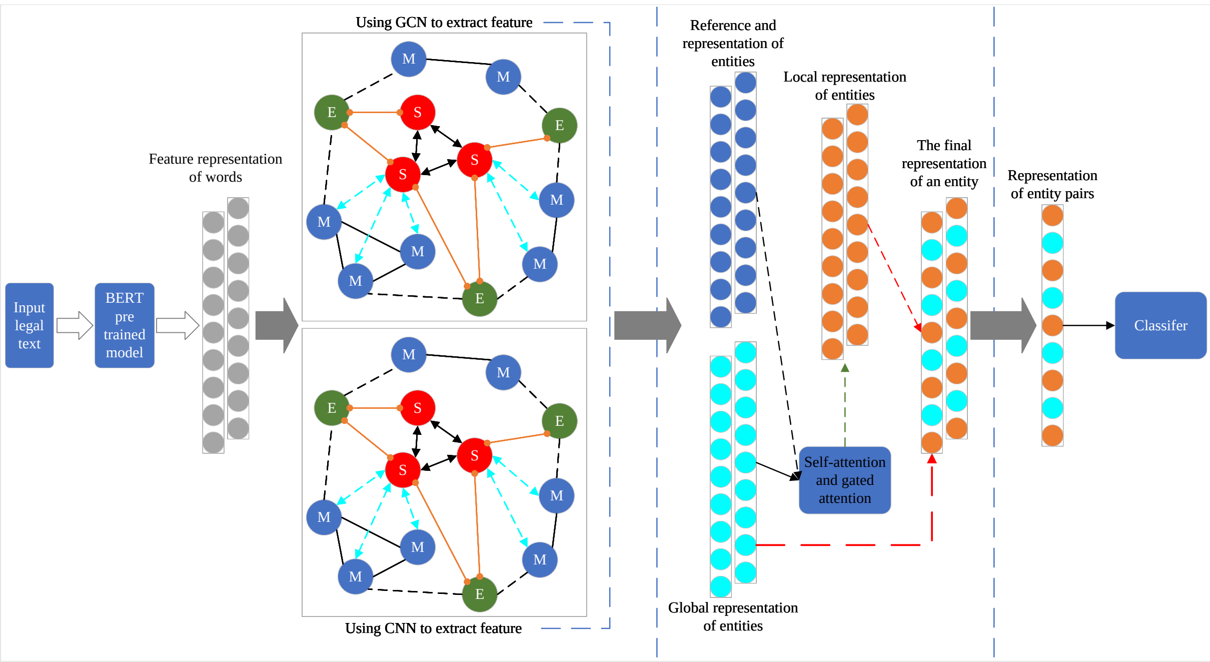

Figure 3. The entire structure of the designed legal text relationship extraction method

Due to the advantage of BERT in word vector modeling, in this paper, we will use BERT model to encode the input legal text g^S. Assuming that legal texts $D^s = [d_1, d_2, ..., d_s]$ and d_s are the input words, the BERT pre-training model is used to encode the word sequences, and finally the encoded word vectors are obtained $C^s = [c_1, c_2, ..., c_s]$. After encoding, the feature representation of each word can be obtained c_s, and a legal text graph is generated using the feature representation of each word. There are three types of nodes in the text graph, namely, word nodes, entity nodes and sentence nodes.

(1) The feature representation of a word node is shown as follow, which is obtained by averaging the vector sum from the start position to the end position.

$$n_m = \left[avg_{c_i \in m}(c_i); r_m\right] \tag{13}$$

where n_m is the feature representation of a word node, $c_i \in m$ refers to the feature representation of a word belonging to a word node, and r_m denotes the feature representation of a word node type.

(2) The feature representation of an entity node is obtained by averaging the feature representations of all the contained word nodes and splicing the feature vectors representing the text type after the feature representation.

$$n_e = \left[avg_{m_i \in e}(m_i); r_e\right] \tag{14}$$

where n_e is the feature representation of the entity node, $m_i \in e$ is the feature representation of all the word nodes belonging to this entity, and r_e is the feature representation of the entity node type.

(3) The feature representation of a sentence node is obtained by averaging the feature representations of all the words contained in a sentence and splicing the feature vectors representing the type of sentence after the feature representation.

$$n_s = \left[avg_{w_i \in s}(c_i); r_s\right] \tag{15}$$

where n_s is the feature representation of the sentence node, $c_i \in s$ is the feature representation of all the words belonging to the current sentence, and r_s is the feature representation of the sentence type node.

4.2. **Legal text feature extraction and enhancement based on graph neural network and attention mechanism.** After obtaining the feature representation of each node in the text graph, the text features are extracted using different feature extraction modules for the graph channel and the sequence channel respectively to mine the text association information. Firstly, GCN is used for feature extraction of entity nodes, then CNN is used for feature extraction of sentence and word nodes, and finally the features of both channels are enhanced using the attention mechanism.

The graph channel adopts the GCN model to form the feature extraction module, and the input sequence is semantically encoded based on location. Then the input graph is constructed to extract the implicit features of the entities using the GCN. The computational process is implied below.

$$F_s = \delta\big(V_o[h_{s-1}, n_e] + b_o\big) \otimes \tanh\big(f_s \otimes J_{s-1} + i_s \otimes J_s^*\big) \tag{16}$$

where V is the weight matrix, b is the bias term, $[h_{s-1}, n_e]$ is the connection between vectors h_{s-1} and n_e, f_s and i_s denote the forgetting gate and input gate, respectively. J_s^* denotes the generated candidate memory cell and J_{s-1} is the long-term memory cell, which are defined as implied in Equation (17) and Equation (18), respectively. δ denotes the Sigmoid function, $\otimes$ denotes the Hadamard product, and the computation rule is

$A \otimes B = (A \otimes B)_{ij} = a_{ij}b_{ij}$, which utilizes hyperbolic tangent to activate J_s and combine with output gate to obtain the final feature F_s.

$$J_s^* = \tanh\big(V_J[h_{s-1}, x_s] + b_J\big) \tag{17}$$

$$J_{s-1} = f_{s-1} \otimes J_{s-2} + i_s \otimes J_s^* \tag{18}$$

The above output is used as the input of the GCN layer to further extract the hidden semantic features of the text. The GCN obtains the feature representation vector of a node through the fusion of the features of the node and the neighboring nodes, and the feature optimization can be further realized through multi-layer iteration. The L-layer GCN graph convolution process is implied in Equation (19).

$$F_i^{(l)} = \delta\left(\sum_{j=1}^{n}(A_{ij}V^{(l)}F_j^{(l-1)} + b^{(l)})\right) \tag{19}$$

where δ denotes the Sigmoid function, V is the weight matrix, b is the bias term, A_{ij} is the adjacency matrix, and finally the final output F^1 of the graph channel can be obtained.

The sequence channel adopts BERT-CNN model to realize the feature extraction of sentence nodes and word nodes, after word embedding with BERT model, CNN is used to average the word vectors, and then splice them with CLS vectors. The result of each convolution is converted into a numeric feature k by using the maximum function, and the numeric features obtained through convolutional pooling are combined into a feature vector to obtain the final fusion feature F^2 of sentence and word.

Subsequently, this paper inputs the output F^1 of the graph channel and the output F^2 of the sequence channel into the cross-attention layer together, and uses the cross-attention mechanism for feature fusion to obtain the feature output. The cross-attention mechanism used can balance the weight allocation of the two channels learning, which can reduce the impact of noise on the vector, thus optimizing the model effect. The self-attention calculation process is implied below.

$$selfAttention(Q, K, H) = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot H \tag{20}$$

where Q is the output of the sequence channel F^2, and K and H are the outputs of the graph channel F^1 and are normalized to maintain gradient stability.

The calculation of the gating attention is indicated below.

$$gateAttention(Q, K, H) = softmax(Q \cdot K^T) \cdot K + (1 - softmax(Q \cdot K^T)) \cdot H \tag{21}$$

where Q is the output of the sequence channel, K is the output of the graph channel, and H is the random initialization vector.

4.3. **Classified output of legal text relationships.** After obtaining the features of the entity, the fusion features of the words and sentences, the F^1, F^2 and the relative distances of the entity can be stitched together as the final representation of the entity by splicing using the cross-attention mechanism described above, as implied below.

$$F_a = \big[F^1; F^2; dis(a, b)\big] \tag{22}$$

$$F_b = \big[F^1; F^2; dis(b, a)\big] \tag{23}$$

where $dis(a, b)$ represents the relative distance from entity a to entity b. Similarly, $dis(b, a)$ represents the relative distance from entity b to entity a.

Ultimately, the final representations of the two entities are stitched together to obtain the final entity pair representation.

$$o_r = [F_a; F_b] \tag{24}$$

Finally, a multiclassification problem is transformed into a binary classification problem using a feed forward neural network to predict the target and contextual relationship representations. r is the predicted probability distribution over the set of all relationships R as indicated below.

$$y_r = sigmod(FFNN(o_r)) \tag{25}$$

where $sigmod$ is used as the distribution function. The loss function is implied below.

$$L = -\sum_{r\in R} \left(y_r^* \log(y_r) + (1 - y_r^*) \log(1 - y_r)\right) \tag{26}$$

5. **Performance testing and analysis.**

5.1. **Comparison experiment.**

This article extracted 2,186 drug-related criminal judgments from the China Judgment and Decision Network (CJDN), covering the period from 2011 to 2020. In this article, there are 960 case description sentences in the training set, containing 3217 pairs of entity relations, and 236 case description sentences in the test set, containing 811 pairs of entity relations. To estimate the efficiency of the suggested method, comparative experiments are conducted in this article. For the convenience of analysis, the method in the literature [17] is denoted as DLBCNN, the method in the literature [20] is denoted as CLHGMF, and the algorithm in this paper is denoted as ATTGNN. The experiments use NVIDIA A100 graphics card, the memory size is 40GB, the deep learning framework uses Pytorch1.7.0, the programming language is python 2.6.3, and the experimental parameters are set as implied in Table 1.

Table 1. Parameters used in the experiment

Parameter	**Dropout**	**Epoch size**	**Batch size**	**Learning rate**	**Decay rate**
value	0.5	50	256	0.01	0.9

In this article, the evaluation metrics Accuracy, Precision, Recall, Macro-F1 [29] and Micro-F1 [30] are used to assess the effectiveness of categorizing legal textual relationships. Table 2 demonstrates the comparative results of each metric for different methods. Table 2 implies that the values of all indicators of the proposed method ATTGNN are better than other methods. It shows the effectiveness of ATTGNN for legal document relationship extraction. Compared with DLBCNN and CLHGMF methods, ATTGNN extracts text features and sentence word features by graph convolutional neural network and CNN, respectively, and can capture complex relationships in sequence data more comprehensively through the advantages of attention and gated attention, thus achieving better performance than a single attentional mechanism in the relationship extraction task. In addition, the Macro-F1 and Micro-F1 of ATTGNN are 0.694 and 0.884, respectively, which are improved by 15.5% and 15.9% compared to the DLBCNN method, and 7.4% and 6.5% compared to the CLHGMF method, thus more validating the effectiveness of the ATTGNN classification.

Table 2. Comparison of classification performance of different law subdivision relation extraction methods

Model	Accuracy	Precision	Recall	Macro-F1	Micro-F1
DLBCNN	0.793	0.778	0.802	0.539	0.725
CLHGMF	0.862	0.841	0.875	0.620	0.819
ATTGNN	0.937	0.922	0.939	0.694	0.884

As can be seen in Figure 4, the suggested ATTGNN model can converge to a lower level using the loss values compared to DLBCNN model and CLHGMF model. More often than not, the loss value tends to stabilize when the model is trained to about the 12th round, while DLBCNN and CLHGMF are still in a fluctuating trend, which shows that the model proposed in this paper can reach the convergence state faster. The DLBCNN method has a higher loss, which is due to the fact that it only extracts the local features of the text by using CNN, which is a larger amount of computation, and it does not take into account the features of the sentences and words, which leads to a higher loss. The CLHGMF method extracts both local and global features of the text using GCN, but uses only a single attentional mechanism for weight allocation, resulting in a higher classification loss than the ATTGNN. The effectiveness of the ATTGNN on the task of relationship extraction in the legal domain is verified by the above performances.

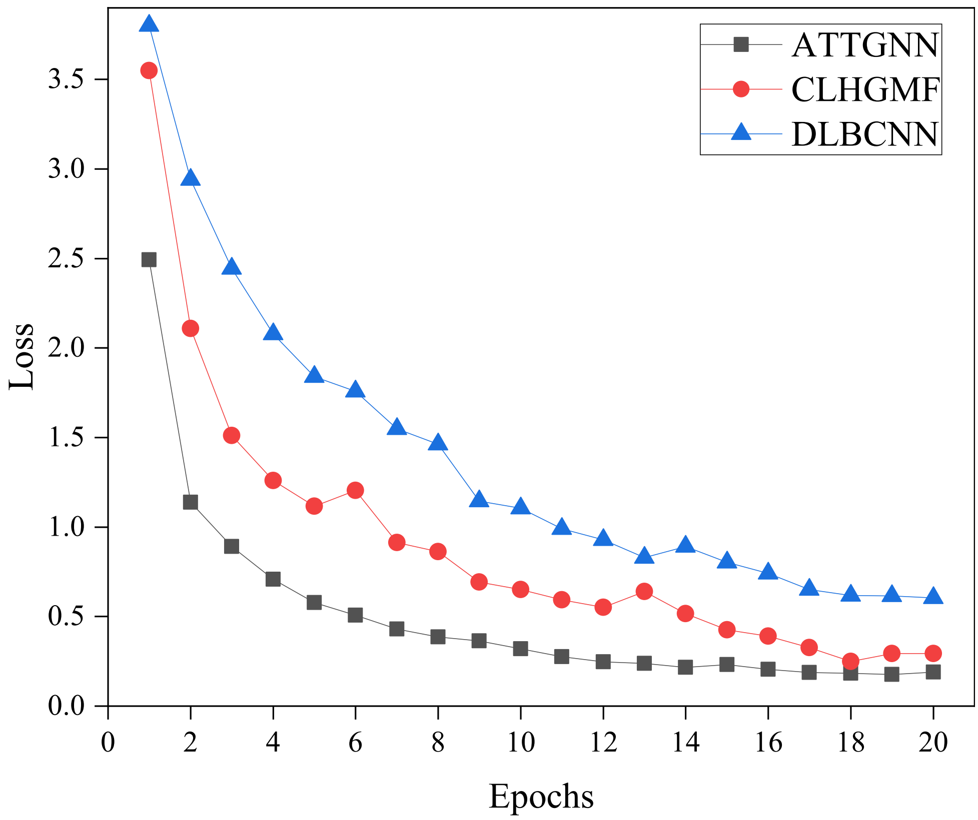

Figure 4. Comparison of loss values of different legal textual relationship extraction methods

5.2. **Ablation experiment.** To verify the role of cross-attention mechanism and graph neural network in ATTGNN in improving the experimental effect, this paper designs five groups of ablation experiments for verification. Experiment 1 Remove the self-attention mechanism, denoted as ATTGNN-SEF. Experiment 2 Remove the gated attention mechanism, denoted as ATTGNN-GRU. Experiment 4 removes the GCN feature extraction module and is notated as ATTGNN-GCN. experiment 5 is the complete model of ATTGNN. The results of the ablation experiments are indicated in Table 3 and Figure 5.

Table 3. Results of ablation experiments of components in the ATTGNN

Method	Accuracy	Precision	Recall	Macro-F1	Micro-F1
ATTGNN-SEF	0.828	0.809	0.815	0.562	0.786
ATTGNN-GRU	0.847	0.824	0.823	0.596	0.801
ATTGNN-GCN	0.783	0.756	0.761	0.489	0.742
ATTGNN-ATT	0.751	0.729	0.710	0.432	0.708
ATTGNN	0.937	0.922	0.939	0.694	0.884

As can be seen from Table 3, the ATTGNN that integrates all the components performs optimally in all the metrics, and the Accuracy of the ATTGNN is 0.937, which is improved by 10.9%, 9%, 15.4%, and 18.6% compared to the ATTGNN-SEF, the ATTGNN-GRU, the ATTGNN-GCN, and the ATTGNN-ATT, respectively.

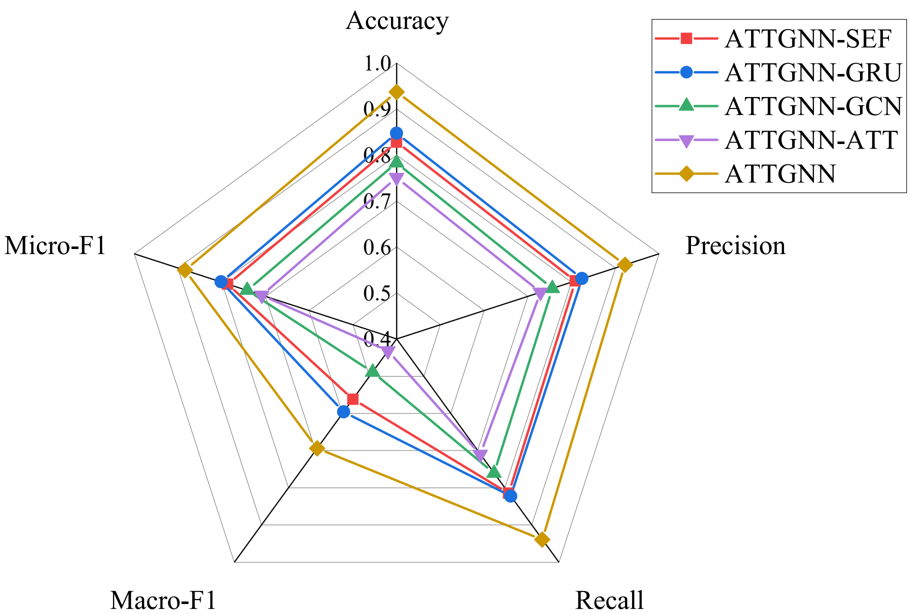

Figure 5. Results of ablation experiments with different components of the ATTGNN

From Figure 5, it can be further found that the extraction effect of using the cross-attention mechanism is greatly improved compared with that of using the single-attention mechanism, which indicates that using the cross-attention mechanism for feature fusion can help the model to obtain more comprehensive text features and enrich the semantic information. Experiment 4 can evaluate the effect of GCN on the model, the experiment proves that if GCN is not used for text feature extraction, the classification performance of ATTGNN will also be reduced, which indicates that the use of GCN can better mine entity relationships, and this phenomenon shows that when the number of layers increases, more information can be exchanged through the attention mechanism, thus improving the performance of the model. Therefore, the ATTGNN with all the components is able to achieve the best performance.

6. **Conclusion.** Focusing on the issue that the current legal text relationship extraction methods have insufficient feature extraction, this article designs a legal text relationship extraction model method relied on attention graph neural network. Firstly, the legal text is represented based on graph structure, so that each legal text corresponds to a separate text graph. Then, the BERT model is used to encode the constructed text graph, so as to better obtain the semantic information of text, sentences and words. Secondly, the legal text features are extracted using dual channels, the graph channel uses GCN for entity node feature extraction, and the sequence channel uses CNN for sentence and word node feature extraction, so as to dig deeper into the text association information. Subsequently, the features of the two channels are enhanced using the self-attention mechanism and gated attention mechanism respectively. Finally, at the classifier layer, the classification output of the legal text entity relationship. Simulation results indicate that the suggested model has further performance improvement over the existing models. The results of the ablation experiment also verify that the different components contribute significantly to the performance enhancement of the proposed model.

REFERENCES

[1] D. Gelbart and J. Smith, "The application of automated text processing techniques to legal text management," *International Review of Law, Computers & Technology*, vol. 8, no. 1, pp. 203–210, 1994.

[2] R. Sil, A. Roy, M. Dasmahapatra, and D. Dhali, "An intelligent approach for automated argument based legal text recognition and summarization using machine learning," *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 5, pp. 5457–5466, 2021.

[3] A. Sleimi, N. Sannier, M. Sabetzadeh, L. Briand, M. Ceci, and J. Dann, "An automated framework for the extraction of semantic legal metadata from legal texts," *Empirical Software Engineering*, vol. 26, pp. 1–50, 2021.

[4] E. Sulis, L. Humphreys, F. Vernero, I. A. Amantea, D. Audrito, and L. Di Caro, "Exploiting co-occurrence networks for classification of implicit inter-relationships in legal texts," *Information Systems*, vol. 106, 101821, 2022.

[5] M.-F. Moens, C. Uyttendaele, and J. Dumortier, "Intelligent information extraction from legal texts," *Information & Communications Technology Law*, vol. 9, no. 1, pp. 17–26, 2000.

[6] F. Solihin, I. Budi, R. F. Aji, and E. Makarim, "Advancement of information extraction use in legal documents," *International Review of Law, Computers & Technology*, vol. 35, no. 3, pp. 322–351, 2021.

[7] Y. Yang, Z. Wu, Y. Yang, S. Lian, F. Guo, and Z. Wang, "A survey of information extraction based on deep learning," *Applied Sciences*, vol. 12, no. 19, 9691, 2022.

[8] B. Hachey and C. Grover, "Extractive summarisation of legal texts," *Artificial Intelligence and Law*, vol. 14, pp. 305–345, 2006.

[9] H. Chen, L. Wu, J. Chen, W. Lu, and J. Ding, "A comparative study of automated legal text classification using random forests and deep learning," *Information Processing & Management*, vol. 59, no. 2, 102798, 2022.

[10] J. Naseri, H. Hassanpour, and A. Ghanbari, "A method for the automatic extraction of keywords in legislative documents using statistical, semantic, and clustering relationships," *International Journal of Nonlinear Analysis and Applications*, vol. 12, no. Special Issue, pp. 265–278, 2021.

[11] A. Thomas and S. Sangeetha, "Semi-supervised, knowledge-integrated pattern learning approach for fact extraction from judicial text," *Expert Systems*, vol. 38, no. 3, pp. e12656, 2021.

[12] V. Tran, M. Le Nguyen, S. Tojo, and K. Satoh, "Encoded summarization: summarizing documents into continuous vector space for legal case retrieval," *Artificial Intelligence and Law*, vol. 28, pp. 441–467, 2020.

[13] Y. Ren, J. Han, Y. Lin, X. Mei, and L. Zhang, "An ontology-based and deep learning-driven method for extracting legal facts from Chinese legal texts," *Electronics*, vol. 11, no. 12, 1821, 2022.

[14] Y.-P. Förster, A. Annibale, L. Gamberi, E. Tzanis, and P. Vivo, "Information retrieval and structural complexity of legal trees," *Journal of Physics: Complexity*, vol. 3, no. 3, 035008, 2022.

[15] Z. Chen, H. Zhang, L. Ye, and S. Li, "An approach based on multilevel convolution for sentence-level element extraction of legal text," *Wireless Communications and Mobile Computing*, vol. 5, pp. 1–12, 2021.

[16] H. Chen, L. Wu, J. Chen, W. Lu, and J. Ding, "A comparative study of automated legal text classification using random forests and deep learning," *Information Processing & Management*, vol. 59, no. 2, 102798, 2022.

[17] M. Gambhir and V. Gupta, "Deep learning-based extractive text summarization with word-level attention mechanism," *Multimedia Tools and Applications*, vol. 81, no. 15, pp. 20829–20852, 2022.

[18] Y. T.-H. Vuong, Q. M. Bui, H.-T. Nguyen, T.-T.-T. Nguyen, V. Tran, X.-H. Phan, K. Satoh, and L.-M. Nguyen, "SM-BERT-CR: a deep learning approach for case law retrieval with supporting model," *Artificial Intelligence and Law*, vol. 31, no. 3, pp. 601–628, 2023.

[19] J. Chen, L. Du, M. Liu, and X. Zhou, "Mulan: A multiple residual article-wise attention network for legal judgment prediction," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 4, pp. 1–15, 2022.

[20] S. Bi, Z. Ali, M. Wang, T. Wu, and G. Qi, "Learning heterogeneous graph embedding for Chinese legal document similarity," *Knowledge-Based Systems*, vol. 250, 109046, 2022.

[21] G. Feng, Y. Qin, R. Huang, and Y. Chen, "Criminal action graph: a semantic representation model of judgement documents for legal charge prediction," *Information Processing & Management*, vol. 60, no. 5, 103421, 2023.

[22] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A Spectral Convolutional Neural Network Model Based on Adaptive Fick's Law for Hyperspectral Image Classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19–46, 2024.

[23] T.-Y. Wu, A. Shao, and J.-S. Pan, "CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm," *Mathematics*, vol. 11, no. 10, 2339, 2023.

[24] T.-Y. Wu, H. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, 1977, 2023.

[25] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[26] M. Ishimaru, Y. Okada, R. Uchiyama, R. Horiguchi, and I. Toyoshima, "Classification of Depression and Its Severity Based on Multiple Audio Features Using a Graphical Convolutional Neural Network," *International Journal of Environmental Research and Public Health*, vol. 20, no. 2, 1588, 2023.

[27] Z. Meng, Z. Wu, and J. Gray, "A collaboration-oriented M2M messaging mechanism for the collaborative automation between machines in future industrial networks," *Sensors*, vol. 17, no. 11, 2694, 2017.

[28] S. Sukharev, M. Betanzos, C.-S. Chiang, and H. R. Guy, "The gating mechanism of the large mechanosensitive channel MscL," *Nature*, vol. 409, no. 6821, pp. 720–724, 2001.

[29] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A novel feature selection algorithm for text categorization," *Expert Systems with Applications*, vol. 33, no. 1, pp. 1–5, 2007.

[30] T. Kang, A. Perotte, Y. Tang, C. Ta, and C. Weng, "UMLS-based data augmentation for natural language processing of clinical research literature," *Journal of the American Medical Informatics Association*, vol. 28, no. 4, pp. 812–823, 2021.