Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

LA-MGFM: A legal judgment prediction method via sememe-enhanced graph neural networks and multi-graph fusion mechanism

Qihui Zhao [a], Tianhan Gao [a], Nan Guo [b,*]

[a] Software College, Northeastern University, 195 Chuangxin Road, Hunnan District, Shenyang 110169, China
[b] School of Computer Science and Engineering, Northeastern University, 195 Chuangxin Road, Hunnan District, Shenyang 110169, China

ARTICLE INFO

Keywords:
Legal judgment prediction
Graph neural networks
Sememe-enhanced gated graph neural networks
Multi-graph fusion mechanism
Text classification

ABSTRACT

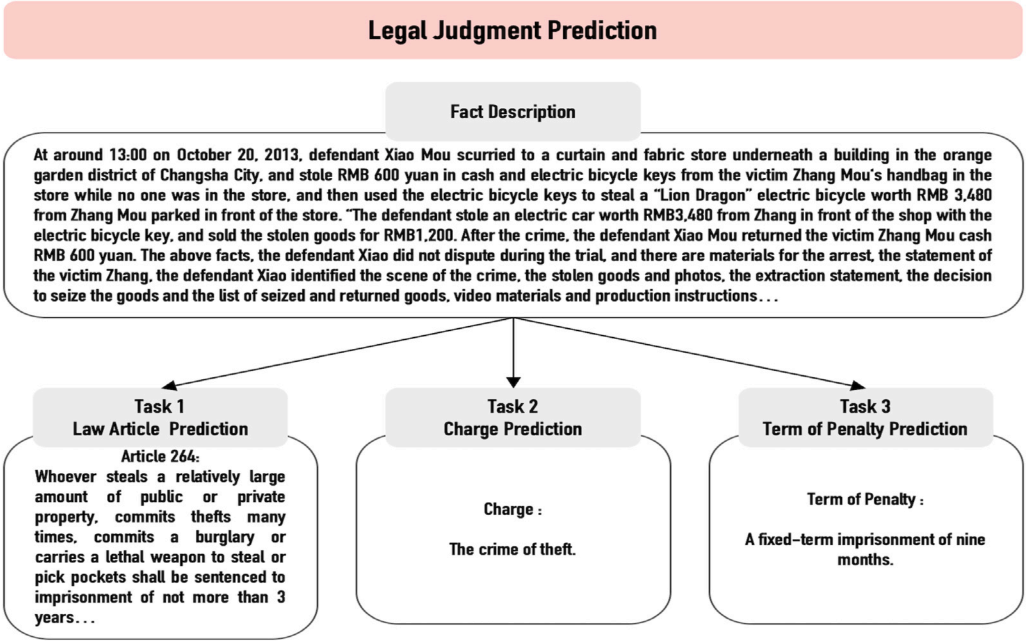
Legal Judgment Prediction (LJP) is a significant task of legal intelligence. Its objective is to predict the relevant law articles, charges, and terms of penalty based on fact descriptions of a criminal case. Existing methods have a drawback: they cannot effectively deal with charges confusion when using various granularity of law articles and predicting outcomes with limited data. In response to this challenge, we propose a solution: a graph neural network-based LJP method that utilizes a multi-graph fusion mechanism to fully and accurately integrate law article information. In detail, we begin by constructing five types of graphs for each case. In the phase of intra-graph information passing, we adopt a Sememe-enhanced Gated Graph Neural Networks to aggregate and update the node features by combining law articles and sememe information. For inter-graph information passing, we introduce a multi-graph fusion mechanism that merges the node features of the five graphs. Finally, we devise a graph readout function, which employs a classifier to derive the results of LJP. The results of our experiment on real-world datasets demonstrate that our method outperforms the current state-of-the-art approaches in our experimental metric.

1. Introduction

Legal Artificial Intelligence (LegalAI) involves various tasks, one of which is Legal Judgment Prediction (LJP). Essentially, LJP aims to obtain the charges, relevant law articles, and terms of penalty based on the fact description of a case. This process can provide prompt professional legal advice to individuals who are not well-versed in the legal field. Due to its significance, LJP has garnered significant attention in recent years and has been recognized as one of the critical tasks in natural language processing. Fig. 1 shows a sample in the LJP dataset, which includes the fact description of the case, the relevant law article, the charge, and the term of penalty. It is difficult to make accurate predictions based solely on the description of a case. This is because there are often subtle differences in the textual descriptions of cases. For example, in the descriptions of the crime of dangerous driving and the crime of traffic casualties, there are “drunk driving causing injury” and “speeding driving causing injury”, respectively. There are only two words in the two sentences that are different, making the model challenging to understand semantically. The legal system in China currently operates under a civil law framework that emphasizes the importance of logical reasoning in jurisprudence. This framework serves as the foundation for judicial trials and mandates that judges strictly adhere to the law articles. These articles play a critical role in determining charges. However, integrating their information into the LJP model presents a challenging task. Fortunately,

[*] Corresponding author.
E-mail addresses: 1910459@stu.neu.edu.cn (Q. Zhao), gaoth@mail.neu.edu.cn (T. Gao), guonan@mail.neu.edu.cn (N. Guo).

https://doi.org/10.1016/j.ipm.2023.103455
Received 2 April 2023; Received in revised form 22 May 2023; Accepted 28 June 2023
Available online 13 July 2023



At around 13:00 on October 20, 2013, defendant Xiao Mou scurried to a curtain and fabric store underneath a building in the orange garden district of Changsha City, and stole RMB 600 yuan in cash and electric bicycle keys from the victim Zhang Mou's handbag in the store while no one was in the store, and then used the electric bicycle keys to steal a "Lion Dragon" electric bicycle worth RMB 3,480 from Zhang Mou parked in front of the store. "The defendant stole an electric car worth RMB3,480 from Zhang in front of the shop with the electric bicycle key, and sold the stolen goods for RMB1,200. After the crime, the defendant Xiao Mou returned the victim Zhang Mou cash RMB 600 yuan. The above facts, the defendant Xiao did not dispute during the trial, and there are materials for the arrest, the statement of the victim Zhang, the defendant Xiao identified the scene of the crime, the stolen goods and photos, the extraction statement, the decision to seize the goods and the list of seized and returned goods, video materials and production instructions...

Article 264:
Whoever steals a relatively large amount of public or private property, commits thefts many times, commits a burglary or carries a lethal weapon to steal or pick pockets shall be sentenced to imprisonment of not more than 3 years...

Charge :

The crime of theft.

Term of Penalty :

A fixed-term imprisonment of nine months.

Fig. 1. An example of dataset (CAIL2018). LJP needs to predict law article, charge, and term of penalty based on a legal document - fact description.

with the advent of deep neural networks, research is underway to address this issue. Luo et al. (2017) first obtains the law articles related to the case fact description by the traditional classifier and then fuses the information of fact description with the relevant law articles to obtain the charges prediction results by the attention mechanism. The traditional classifier, on the other hand, introduces too much noise to the model and shows average performance in predicting confusing charges. Hu et al. (2018a) introduces ten discriminative legal attributes to assist in charge prediction. Although the method has some performance enhancement, the method relies on domain knowledge of the legal expertise. People prefer methods capable of automatically extracting useful features in practical scenarios. Xu et al. (2020a) constructs a graph from law articles and incorporates the information from these articles into the feature extraction process of case text through an attention mechanism. However, a drawback of this method is that the graph construction process for the law articles is relatively simplistic, making it difficult to gather enough information for the LJP task.

In this paper, we present a novel graph neural network-based method to address the issue mentioned earlier. To start, we use the LA Critical Feature Extractor to obtain K law article communities representation. Additionally, we construct five kinds of text graphs based on the fact descriptions. We then use a GRU-based GNN for feature extraction of the five graphs. We also introduce a multi-graph fusion mechanism to combine the node features of the five graphs and facilitate a readout after inter-graph message passing. Finally, we utilize a softmax classifier for the LJP. The contributions to our paper's work include the following.

1. Five kinds of text graphs are constructed for the legal judgment prediction task, and a Sememe-enhanced GGNN is proposed to incorporate the law articles and Sememe information.
2. A multi-graph fusion mechanism is proposed to solve the information fusion between different graphs.
3. The confusion problem of charges is effectively alleviated. Better results are obtained than SOTA model on real-world dataset and the good performance is guaranteed on the dataset with only a small number of samples.

2. Related work

2.1. Legal judgment prediction

In recent years, LJP has been regarded as being synonymous with text classification, a classic and essential task in natural language processing (Minaee et al., 2021). Typically, LJP is classified into four different types. The first type is based on mathematical and quantitative analysis (Kort, 1957; Nagel, 1964) and is most often applied to datasets with a limited amount of data available. In early work, linear classifiers were frequently used and legal rules were incorporated to ensure interpretability, but they were generally not good at generalization. Meanwhile, the second approach involved manual feature engineering and employed traditional machine learning algorithms to generate prediction results. Liu and Hsieh (2006) took a different route and utilized word-level and paragraph-level feature information in their K-nearest neighbor algorithm (KNN) to predict charges. Liu et al. (2015) utilized a support vector machine (SVM) to obtain initial stage classification results, which were subsequently re-ranked using word-level features and word co-occurrence frequency information to arrive at final charges classification results. In order to predict charges, Katz et al. (2014) utilized case-based information like location and time as feature information. However, this approach relies on manual shallow feature extraction, which not only necessitates legal professionals' aid but is also inadequate for completely extracting semantic features from legal documents at varying levels. The third type of methods focuses on improving LJP performance by introducing new structures. Wang et al. (2019) developed a model that utilizes a combination of FastText and TextCNN to solve the LJP. Liu et al. (2019) proposed the HLCP model, which is an attention-based end-to-end model that predicts charges. Jiang et al. (2018) presented an approach that combines neural networks with reinforcement learning to tackle charge prediction and enhance the model's interpretability. Long et al. (2019) converted the charge prediction task from a general text classification task to a machine reading comprehension task, demonstrating excellent performance. Li et al. (2019) presented a multichannel attentive neural network model for extracting essential information about the case, the defendant's profile, and law articles. Chen et al. (2019) leveraged a gate mechanism-based model to enhance penalty prediction accuracy. Pan et al. (2019) developed an attention-based multilevel architecture that can handle cases with multiple defendants. The fourth approach involves examining how legal knowledge can be integrated into the model. Kang et al. (2019) presented a method for extracting the relevant information from the factual description by analyzing the definition of a charge. This method improved the feature representation of the factual description and subsequently increased the accuracy of the charge prediction. Zhong et al. (2018) developed a charge prediction system using the directed acyclic graph (DAG) theory to establish the topological dependencies of LJP's subtasks. With the incorporation of external knowledge, Wei and Lin (2019) validated the efficacy of the model. Moreover, Hu et al. (2018b) identified 10 critical legal attributes and Xu et al. (2020b) utilized a graph-based approach to amalgamate legal information into the charge prediction model, resulting in improvements.

2.2. Graph neural networks

Graph Neural Networks (GNN) have garnered significant attention owing to their exceptional computational prowess. While Long Short Term Memory Network (LSTM) and Convolutional Neural Network (CNN) excel in processing Euclidean spatial data (language, image, video, etc.), they face limitations when dealing with non-Euclidean spatial data (social networks, information networks, etc.). To overcome this challenge, researchers leverage graph theory's abstract representation of structured non-Euclidean data by introducing graphs. Graph neural networks are a powerful tool for analyzing graph data by thoroughly examining its characteristics and patterns. The methods proposed by Kipf and Welling (2017), Hamilton et al. (2017), and Gilmer et al. (2017) all use different techniques to aggregate the information from the neighbors and generate a feature vector representation of the nodes. Kipf and Welling (2017) proposed graph convolutional networks, which employ average pooling to aggregate the neighbor nodes. Hamilton et al. (2017) introduced a unified message passing framework referred to as MPNN, while Gilmer et al. (2017) proposed the GraphSAGE approach, which enables different operations, such as averaging or LSTM, to aggregate the neighbor information.

This paper focuses on the GNN-based method for text classification, which involves two main components: graph construction and graph representation. The first step involves constructing a document graph to capture the complex relationships between nodes. Next, a GNN is applied to the learned node-level embeddings, which are subsequently used in the softmax layer for classification purposes. Ultimately, this approach offers a powerful way to classify text, leveraging the rich information present in the document graph. In constructing graphs, the initial focus has been on Semi-supervised text classification, which utilizes a small quantity of labeled data and a large amount of unlabeled data for training. As of late, several semi-supervised techniques based on GNN (Hu et al., 2019; Liu et al., 2020; Yao et al., 2019) have been proposed for text classification, with the goal of more accurately representing the inherent links between the words and documents in a corpus. Overall, these approaches involve creating a heterogeneous graph for the entire corpus that incorporates both word and document nodes. Edge weights are typically determined by assessing the co-occurrence of words and the connections between words and documents (Liu et al., 2020; Yao et al., 2019). Hu et al. (2019) suggests the enrichment of document semantics with supplementary information (i.e., topics and entities). This will involve constructing a heterogeneous information network that includes document, topic, and entity nodes based on predefined rules. The edges between these nodes will contain three types: document-topic, document-entity and entity-entity. However, a downside of semi-supervised text classification is its inability to handle unseen documents during the testing phase. To tackle this issue, certain GNN-based methods (Defferrard et al., 2016; Huang et al., 2019; Zhang et al., 2020a) suggest constructing a distinct graph of words for each document using word similarity or co-occurrence between words within a fixed-size contextual window. Unlike static graphs, dynamic graphs do not depend on domain-specific prior knowledge, and their structure can be acquired concurrently with other learning modules of the system. Chen et al. (2020) propose to view every word in a text as a node in a graph and constructing a separate graph for each document in a dynamic manner. After constructing the document graph, the next important step is graph representation learning. Early graph-based text classification models attempted to extend the functionality of CNNs to incorporate graph CNNs and represent graph-structured text directly (Defferrard et al., 2016). As research on Graph Neural Networks (GNNs) continues to advance, recent studies have focused on exploring different types of GNN models for text classification, such as GCN (Chen et al., 2020; Yao et al., 2019), GGNN (Zhang et al., 2020a), and MPM (Huang et al., 2019). Liu et al. (2020) proposes a novel approach called TensorGCN. This method involves performing intra-graph convolutional propagation followed by inter-graph convolutional propagation. Hu et al. (2019) proposes a heterogeneous graph attention network that is based on a unique Bi-level attention mechanism, which comprises both node-level and type-level attention. In our research, we introduce a dynamic graph-based approach for graph construction and propose a Sememe-Enhanced GGNN for graph representation learning.

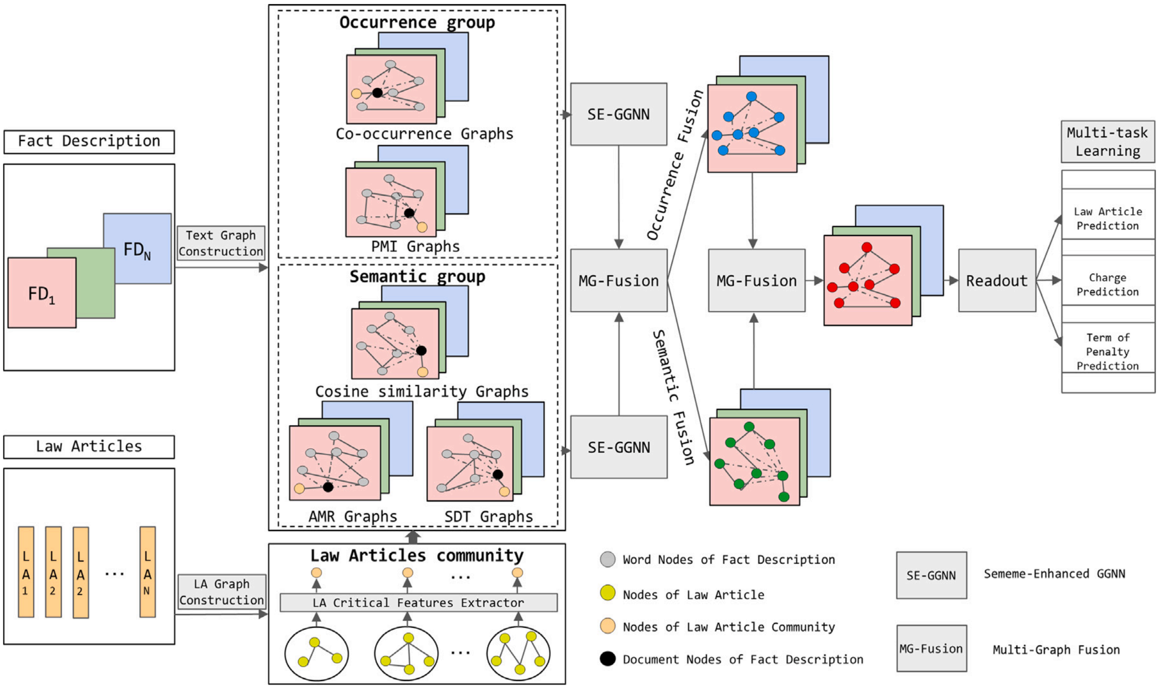


Fig. 2. The proposed LJP architecture: LA-MGFM. LA Critical Features Extractor can extract each law article community's feature representation. SE-GGNN is Sememe-Enhanced Gated Graph Neural Networks. MG-Fusion is Multi-Graph Fusion.

3. Problem formulation

In this section, we briefly describe the fundamental definitions in this work, such as fact description, law article, charge, term of penalty, and legal judgment prediction.

- **Fact description**: Fact description is a part of the judgement document, which mainly describes crime acts, crime times, crime locations, crime consequences, crime tools, and judgment made by court. We note the fact description as FD.
- **Law article**: Law article is the basis for convictions, where each charge has at least one law article. We note the law article as T_l.
- **Charge**: Charge is the crime definition that is contained in the Criminal Law of the People's Republic of China, such as the crimes of theft, fraud, and forcible seizure, etc. We note the charge as T_c.
- **Term of penalty**: Term of penalty is the term used in current law articles for penalties that deprive criminals of their personal freedom. We note term of penalty as T_t.
- **Legal Judgment Prediction (LJP)** : LJP resembles text classification task, which aims to predict the judgment result by exploring the input legal document. As shown in Fig. 1, the input of LJP is a fact description, and the output of LJP contains three results: law articles, charges, and terms of penalty, which comes from the prediction results of three subtasks in LJP. And we seek to train a LJP model M to predict judgment results. In this paper, the LJP model can be formalized as $M(FD) = \{T_c, T_l, T_t\}$.

4. Our method

In this section, we present a novel framework for LJP in detail. Fig. 2 shows our LJP method's architecture (LA-MGFM). Firstly, we construct K law article communities and then using a LA critical features extractor to get the representations of each community. Then the fact descriptions and corresponding law article community are converted into five kinds of text-level graphs. The five textual graphs' nodes features are further aggregated and updated through a Sememe-Enhanced gated graph neural networks (SE-GGNN). Afterward, we fuse feature information of the five kinds of graphs through a multi-graph fusion mechanism (MG-Fusion). Finally, the merged node representations are input to the classifier to obtain the law articles, charges, and terms of penalty. Our method's architecture consists of five parts: law article community, graph construction, intra-graph word interaction, multi-graph fusion mechanism, and multi-task learning (prediction).

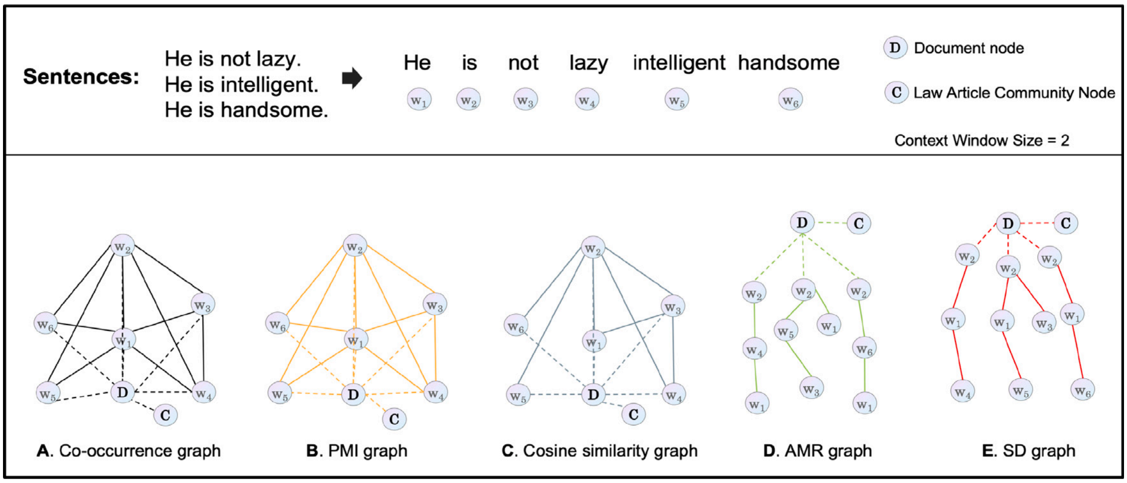


Fig. 3. Five examples of text graph.

Table 1
Graph construction details.

	Co-occurrence graph	PMI graph	AMR graph	Cosine similarity graph	SD graph
Node Category	1. word nodes; 2. document nodes; 3. law article community nodes				
Node initial vector	word nodes: average word embedding of LawFormer document nodes: [CLS] embedding of LawFormer law article community nodes: See Section 4.1.2				
Edge Category	1. word-word; 2. document-word; 3. law article community-document				
Edge Initial Vector					
word-word	Eq. (5)	Eq. (6)	Eq. (9)	Eq. (8)	Eq. (10)
document-word	TF-IDF				
law article community-document	random number that follows a normal distribution				

4.1. Graph construction

Our goal is to effectively model text graphs that incorporate both semantic and local contextual information. Additionally, we can enhance the performance of LJP by utilizing several graphs at varying levels of detail. By using multiple graphs, our model gains a more comprehensive understanding of node representations and learns more valuable information.

4.1.1. Overall settings and notations

We construct five categories of text graphs as $\{G_s = (V, E_n)\}_{n=1}^{N}$ based on the fact description and relevant law article community. V and E_n V represent the node set and edge set. $N = 5$ implies we construct five kinds of text graph. $V = \{V_w, V_{fd}, V_{la}\}$, where V_w, V_d, and V_{la} are word nodes (text of fact description), document nodes and law article community nodes. Specifically, document nodes are linked to all word nodes of the fact description, while law article community nodes are linked to the document nodes. We begin by constructing text nodes using a standard text preprocessing technique. This involves dividing the text into individual words, filtering out any stop words, and utilizing the remaining words as nodes within our co-occurrence graphs, point-wise mutual information graphs, and cosine similarity graphs. However, when it comes to constructing abstract meaning representation graphs and semantic dependency graph graphs, we input the entire sentence directly into the tool. The tool outputs both node and edge information for these graphs. We construct three types of edges: word-document edges, document-law article community edges, and word-word edges. Word-document edges are produced using word co-occurrence in documents, with edge weights determined by the term frequency-inverse document frequency (TF-IDF). For document-law article community edges, edge weights are assigned random numbers drawn from a normal distribution. In all graphs, word-document and law article-document edges are established and measured according to the aforementioned rules. In our method, we build word-word edges by considering two groups of properties: local contextual and semantic. Using these five different kinds of word-word edges, we construct five text graphs to represent text documents, where e_{ij} ($e_{ij} \in E_n$) denotes the relationship between word i and word j, as explained in the following subsection. The five text graphs are categorized into the local contextual information group and the semantic information group. Table 1 provides a summary of the construction details of the text graph, including the various types of nodes and their initial vectors, as well as the types of edges and their initial vectors for each text graph. Additionally, Fig. 3 displays examples corresponding to the five different text graph types.

We utilize the pre-trained model LawFormer (Xiao et al., 2021) for the initial vector of the word nodes. Furthermore, we initialize the feature of the word nodes using the parameters of the second-to-last layer in LawFormer. The corresponding equation is as follows:

$$\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_k] = \textbf{LawFormer}([w_1, w_2, \dots, w_k]) \tag{1}$$

where w_k is the word of the input text, $\mathbf{h}_k$ is the output of the second-to-last layer of LawFormer with dimension of $\mathbb{R}^{d_w}$. Additionally, if the text length surpasses the model's threshold, we implement segmented input. Finally, for both the document node and law article nodes, we use the embedding of "[CLS]" from each input as the initial embedding.

4.1.2. Representations of law article community nodes

Inspired by Xu et al. (2020b), we aim to obtain a representation of each law article community by calculating the cosine similarity between them. This is based on their respective TF-IDF values. We then establish a threshold value t to determine which pairs of law articles to include in our analysis. Any pairs with a similarity score equal to or greater than t are retained. From these pairs, we then identified a total of M unconnected subgraphs, which we refer to as communities, specifically represented by $c_1, \dots, c_K$. In order to extract discernible information from each community, we have designed the LA Critical Feature Extractor. This functions on the principle of learning effective key information features that are truly distinguishable, by eliminating similar features between nodes. When processing an arbitrary legal article LA_i, the extractor utilizes a trainable weight matrix W_s to identify similar information between that article LA_i and its neighbors within graph C. As each layer is applied, the similar feature information is removed from the representation of LA_i and its neighboring nodes.

$$\mathbf{h}_{LA_i}^{(l+1)} = W_s^{(l)} \mathbf{h}_{LA_i}^{(l)} - \sum_{LA_j \in N_i} \frac{W_{d_1}^{(l)} \mathbf{h}_{LA_i}^{(l)} + W_{d_2}^{(l)} \mathbf{h}_{LA_j}^{(l)} + \mathbf{b}_d^{(l)}}{|N_i|} \tag{2}$$

where $\mathbf{h}_{LA_i}^{(l)} \in \mathbb{R}$ refers to the representation of law article LA_i in the l the LA Critical Feature Extractor layer, N_i refers to the neighbor set of LA_i in graph G, $\mathbf{b}_d^{(l)}$ is the bias, and $W_s^{(l)} \in \mathbb{R}^{d_l * d_l}$ and $W_{d_2}, W_{d_2} \in \mathbb{R}^{d_l * d_l}$ are the trainable weighted matrix and the neighbor similarity extracting matrix respectively. d_l is the dimension of the feature vector in the lth layer.

We utilize the final layer of the extractor to serve as a representation of the law articles, $\mathbf{h}_{LA_i}^{(l)} \in \mathbb{R}^{d_L}$. This particular layer contains a diverse set of key features that are integral to understanding the law articles. However, to enhance the law articles' distinct features, we proceed to calculate their distinguishing vector $\mathbf{v}_{c_i}$. This computation involves utilizing set operations to combine the key features of the law articles in each community, as shown in Eq. (2). Specifically, we use the set operation of averaging among elements, denoted by $AvP(-)$.

$$\mathbf{h}_{c_i} = \left[\mathrm{AvP} \left(\left\{ \mathbf{v}_{LA_i}^{(L)} \right\}_{LA_j \in g_i} \right) \right] \tag{3}$$

In order to incorporate the law article community nodes into the text graph, we develop a function that predicts the most relevant communities to a given document. The function, which utilizes the basic representation of the fact description f as $\mathbf{h}_f$, relies on a trainable weight matrix $\mathbf{W}_g$ and bias $\mathbf{b}_g$.

$$\hat{\mathbf{X}} = \mathrm{softmax} \left(\mathbf{W}_g \mathbf{h}_f + \mathbf{b}_g \right) \tag{4}$$

The most relevant community c_i is computed as $\hat{c} = \arg\max_{i=1,\dots,K} \hat{X}_i$, where each element $X_i \in \hat{\mathbf{X}}, i = 1, \dots, K$ reflects the degree of correlation between the fact description f and the law article community c_i. Finally, we use the key information vector $\mathbf{h}_c$ of the corresponding community to construct the initialization vector of the law article community nodes in the text graph.

4.1.3. Local contextual information group

The group dedicated to local contextual information is comprised of two types of text graphs: the co-occurrence graph and the point-wise mutual information graph. Local contextual information pertains to the linguistic characteristic of the proximity of words to each other and is regularly utilized in text representation learning (Huang et al., 2019; Yao et al., 2019).

Co-occurrence Graph: Previous models, namely word2vec (Tomás Mikolov et al., 2013), have utilized co-occurrence information to aid in their modeling approach. These models have produced word vectors that have been widely applied in natural language processing with successful outcomes in various downstream tasks. As such, it can be concluded that co-occurrence information is crucial in text modeling. The word co-occurrence phenomenon refers to the concept that words are likely to appear together within a fixed sliding window. In order to create this graph, we first determine a set sliding window size. Next, we calculate the weights of the edges that connect each pair of words based on their number of co-occurrences. If there were no co-occurrences between two words, then there will be no edge present. The value of each entry in the co-occurrence matrix represents the maximum likelihood estimation (MLE) of a given word pair (w_i, w_j), assuming that w_i appears. This estimation can be expressed by the following equation.

$$e_{ij} = p(w_j | w_i) = \frac{\#N(w_i, w_j)}{\#N(w_i)} \tag{5}$$

where $\#N(w_i)$ denotes the number of occurrences of the word w_i, while $\#N(w_i, w_j)$ represents the frequency of the word pair (w_i, w_j) co-occurring in a fixed window. When $\#N(w_i, w_j)$ is greater than 0, it indicates the presence of a co-occurrence relationship between the two words.

Point-Wise Mutual Information Graph: Point-Wise Mutual Information (PMI) is a technique for computing word association, which measures how closely related two words are. The PMI value is determined by calculating the probability of two words appearing together in the text, with a higher value indicating a stronger relevance and greater association. To compute the PMI value for w_i and w_j, use the following formula.

$$\begin{aligned} e_{ij} &= PMI(i,j) = log\frac{p(i,j)}{p(i)p(j)} \\ p(i,j) &= \frac{\#N(w_i,w_j)}{\#N(windows)} \\ p(i) &= \frac{\#N(w_i)}{\#N(windows)} \end{aligned} \tag{6}$$

where $\#N(windows)$ refers to the number of sliding windows, and $\#N(w_i,w_j)$ denotes how many windows contain both w_i and w_j. Meanwhile, $\#N(w_i)$ represents the number of windows containing w_i. We can interpret a PMI value greater than 0 as indicating a higher semantic correlation between two words - the larger the value, the stronger the correlation. Conversely, if the PMI value is less than 0, the two words can be considered unrelated. We assign edges to word pairs with PMI values greater than 0, and disregard those with values less than or equal to 0.

4.1.4. Semantic information group

The semantic information group is made up of three different types of text graphs: the cosine similarity graph, the abstract meaning representation graph, and the semantic dependency graph. Semantic information is crucial to most natural language processing tasks and must be carefully considered. A strong semantic representation is able to capture the implicit rules of language as well as the shared knowledge and understanding that is present in the text (Shervin et al., 2021).

Cosine Similarity Graph: A practical method for measuring semantic relationships between words is through the use of pre-trained word embedding vectors. To extract knowledge from these vectors and measure the relationship between pairs of words, we have chosen to use the cosine similarity measure, a widely used metric in NLP tasks. Specifically, we adopt the Cosine Similarity metric of the word pair as follow:

$$COS(w_i,w_j) = \frac{E_{w_i}\cdot E_{w_j}}{|E_{w_i}|\cdot|E_{w_j}|} \tag{7}$$

where E_{w_i} and E_{w_j} is the embedding of word w_i and w_j.

In the process of constructing a cosine similarity graph, pairs of words are deemed semantically relevant if their cosine similarity value exceeds a threshold of p. Conversely, pairs of words with cosine similarity values less than p are considered unrelated. The edge weight between words is determined using the following calculation:

$$e_{ij} = \frac{\#N_{cos}(w_i,w_j)}{\#N_t(w_i,w_j)} \tag{8}$$

where $\#N_{cos}(w_i,w_j)$ represents the number of times that a pair of words is connected by a semantic relationship across the entire document, while $\#N_t(w_i,w_j)$ represents the number of times that the same pair of words co-occur within a single sentence. If the relevance between the pair of words is 0, then we can conclude that there is no edge connecting them.

Abstract Meaning Representation Graph: To gather sufficient semantic information for LJP, we create an AMR graph for every document. Initially, we extract the sentence-level AMR graphs through the AMR parsing model. Although the edge in the AMR graph is directed, we consider it as an undirected edge. We then tally the frequency of word pairs that have AMR across the entire corpus, and utilize this data to calculate the weight of each pair's edge.

$$e_{ij} = \frac{\#N_{amr}(w_i,w_j)}{\#N_t(w_i,w_j)} \tag{9}$$

$\#N_{amr}(w_i,w_j)$ is defined as the frequency with which a pair of words has an edge in the sentence-level AMR graph across the entire document. For any words that are not present in the graph, we assign a node embedding of 0 and an edge weight of 0.01.

Semantic Dependency Graph: We utilize a semantic dependency graph (tree) to extract sentence semantics. These graphs do not necessarily abstract the vocabulary, but rather describe it through its semantic frame. Semantic dependency parsing seeks to go beyond the superficial syntactic structure of a sentence and obtain direct access to its more profound semantic information. Like the AMR graph, we also connect the graph's root node to the document node to create a document-level semantic dependency graph. To generate the semantic dependency tree, we rely on DDParser (Zhang et al., 2020b) as our tool. The equation provided below is used to calculate the weights of the edges for the two word nodes in the graph:

$$e_{ij} = \frac{\#N_{sd}(w_i,w_j)}{\#N_t(w_i,w_j)} \tag{10}$$

where $\#N_{sd}(w_i,w_j)$ represents the frequency of a specific occurrence. Specifically, it counts the number of times two words in a pair have an edge in a semantic dependency tree across the entire document.

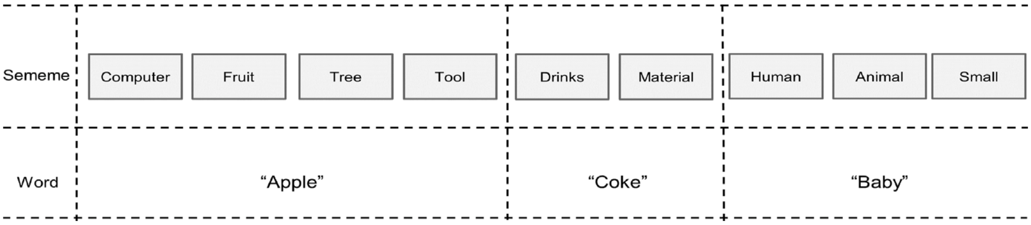


Fig. 4. Examples of sememe information of words.

4.2. Intra-graph word interaction

Once the text graphs have been constructed, the crucial next step is determining how to execute intra-graph interactions. This is where our Sememe-Enhanced Gated Graph Neural Networks (SE-GGNN) design comes into play. Our motivation behind this approach is twofold. Firstly, we utilize GGNN-based graph neural networks to enhance the ability to disseminate information within graph structures over time. Secondly, since text graphs are constructed based on individual words, we integrate fine-grained knowledge (sememe) of words to intensify the efficacy of intra-graph word interactions.

Standard GGNN: The standard GGNN can be expressed as Eqs. (11)–(16), where $\boldsymbol{h}_v^{(1)}$ denotes the initial hidden vector of node v. $\boldsymbol{A}_{v:}$ constitutes the two columns that correspond to node v from the A matrix. $\boldsymbol{a}_v^{(t)}$ represents the outcome of the interaction between the current node and its neighboring nodes through edges, taking information transfer in both directions into account. Eqs. (13)–(16) implement a computation process similar to that of the GRU model. In this process, $\boldsymbol{z}_v^t$ controls the forgotten information and $\boldsymbol{r}_v^t$ controls the newly generated information. The first half of Eq. (16) selects which past information should be disregarded, while the second half selects which newly generated information should be retained. The resulting updated node representation is denoted by $\boldsymbol{h}_v^{(t)}$.

$$\boldsymbol{h}_v^{(1)} = \left[\boldsymbol{x}_v^T, \boldsymbol{0}\right]^T \tag{11}$$

$$\boldsymbol{a}_v^{(t)} = \boldsymbol{A}_{v:}^T \left[\boldsymbol{h}_1^{(t-1)T} \cdots \boldsymbol{h}_{|\mathcal{V}|}^{(t-1)T}\right]^T + \boldsymbol{b} \tag{12}$$

$$\boldsymbol{z}_v^t = \sigma\left(\boldsymbol{W}^z \boldsymbol{a}_v^{(t)} + \boldsymbol{U}^z \boldsymbol{h}_v^{(t-1)}\right) \tag{13}$$

$$\boldsymbol{r}_v^t = \sigma\left(\boldsymbol{W}^r \boldsymbol{a}_v^{(t)} + \boldsymbol{U}^r \boldsymbol{h}_v^{(t-1)}\right) \tag{14}$$

$$\widetilde{\boldsymbol{h}_v^{(t)}} = \tanh\left(\boldsymbol{W} \boldsymbol{a}_v^{(t)} + \boldsymbol{U}\left(\boldsymbol{r}_v^t \odot \boldsymbol{h}_v^{(t-1)}\right)\right) \tag{15}$$

$$\boldsymbol{h}_v^{(t)} = \left(1 - \boldsymbol{z}_v^t\right) \odot \boldsymbol{h}_v^{(t-1)} + \boldsymbol{z}_v^t \odot \widetilde{\boldsymbol{h}_v^{(t)}} \tag{16}$$

Sememe-enhanced GGNN: We introduce the integration of sememe knowledge to provide a more detailed and refined lexical representation of words. By combining various sememe information, we aim to enhance the granularity of word analysis. Sememe theory suggests that a word can be defined by multiple sememes. Fig. 4 provides a few examples to demonstrate this concept. We utilize HowNet (Dong & Dong, 2003; Qi et al., 2019) to obtain sememe information for all words and integrate the operation of incorporating sememe knowledge into the fundamental steps of GGNN. Taking inspiration from Qin et al. (2020) and Zhao et al. (2022), we propose the integration of a sememe cell to regulate sememe knowledge integration. To begin, we obtain the initialized embedding of the corresponding sememe knowledge using a pre-trained model named SAT (Niu et al., 2017). Furthermore, we assign sememe embeddings of 0 to words lacking sememe information. The Sememe-enhanced cell's equations are presented below:

$$\begin{aligned}
\boldsymbol{s}^t &= \frac{1}{n} \sum_{i \in n} \boldsymbol{s}_i \\
z_s^t &= \sigma(W_z[\boldsymbol{s}^t; \boldsymbol{h}_s^{t-1}] + b_z) \\
r_s^t &= \sigma(W_r[\boldsymbol{s}^t; \boldsymbol{h}_s^{t-1}] + b_r) \\
\tilde{\boldsymbol{h}}_v^t &= \tanh(W_h r_s^t * [\boldsymbol{h}_s^{t-1}] + b_h) \\
\boldsymbol{h}_{vs}^t &= (1 - z_s^t) \otimes \boldsymbol{h}_v^{t-1} + z_s^t \otimes \tilde{\boldsymbol{h}}_v^t
\end{aligned} \tag{17}$$

where n represents the number of sememes in a word, $\boldsymbol{s}^t$ represents the embedding of word i in sememe space, z_s^t and r_s^t are the reset and update gates, respectively, and W_z, W_h and W_r are trainable parameters. Additionally, $\boldsymbol{h}_{vs}^t$ represents the hidden state of the sememe cell at time step t.

In order to incorporate the sememe cell into the standard GGNN process, we can utilize the following equation:

$$\begin{aligned}
\boldsymbol{h}_v^1 &= [h_v^\top, 0]^\top \\
a_v^t &= [(\boldsymbol{h}_1^{t-1})^\top, \ldots, (\boldsymbol{h}_{|V|}^{t-1})^\top] A_{v:}^\top + b \\
z_v^t &= \sigma(W_z^1 a_v^t + \boldsymbol{U}^z \boldsymbol{h}_v^{t-1} + \boldsymbol{T}^z \boldsymbol{h}_{vs}^{t-1})
\end{aligned}$$

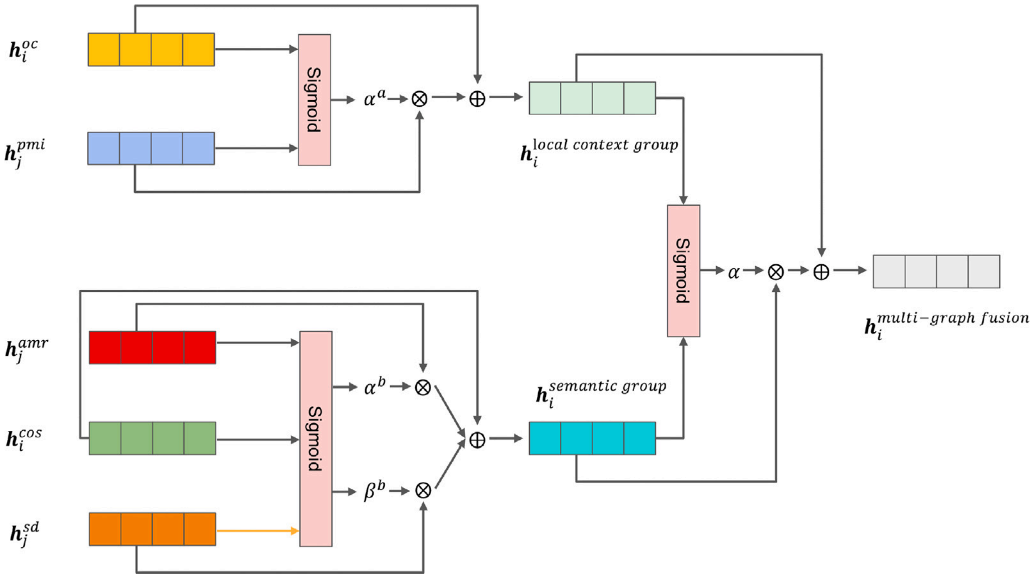


Fig. 5. Our multi-graph fusion mechanism, where $\boldsymbol{h}_i^{oc}$, $\boldsymbol{h}_i^{cos}$, represent the ith node's vector of co-occurrence graph, cosine similarity graph. $\boldsymbol{h}_j^{pmi}$, $\boldsymbol{h}_j^{amr}$, $\boldsymbol{h}_j^{sd}$ represent the jth node's vector of PMI graph, AMR graph and semantic dependency graph, respectively. $\boldsymbol{h}_i^{multi-graphfusion}$ is the vector after multi-graph fusion. ⊗ is Kronecker product and ⊕ is direct sum.

$$
\begin{aligned}
r_v^t &= \sigma(W_r^1 a_v^t + U^r \boldsymbol{h}_v^{t-1} + T^r \boldsymbol{h}_{vs}^{t-1}) \\
\hat{\boldsymbol{h}}_v^t &= \tanh(W_a a_v^t + U(r_v^t \otimes (\boldsymbol{h}_v^{t-1} + \boldsymbol{h}_{vs}^{t-1}))) \\
\boldsymbol{h}_v^t &= (1 - z_v^t) \otimes \boldsymbol{h}_v^{t-1} + z_v^t \otimes \hat{\boldsymbol{h}}_v^t
\end{aligned}
\tag{18}
$$

where the initial vector of nodes v is represented by $\boldsymbol{h}_v^1$. When the dimension is lower than the set value, it is succeeded by a 0. The concatenation of all node features at time $t-1$ is denoted by $[(\boldsymbol{h}_1^{t-1})^\top, \dots, (\boldsymbol{h}_{|V|}^{t-1})^\top]$. $A_{v:}$ is determined by selecting from the graph adjacency matrix A. a_v^t represents the outcome of the interaction between nodes and their adjacent contacts through edges. There are three controls in place: $1 - z_v^t$ controls forgetting information, z_v^t controls remembering new information, and r_v^t controls which previously generated information is used for new information. The operation ⊗ is a Hadamard element-wise product, and $\hat{\boldsymbol{h}}_v^t$ represents newly generated information. Finally, after t iterations of GGNN, the resulting final updated node vector representation of node v is represented by the $\boldsymbol{h}_v^t$.

After one iteration, we take a step to enhance the interaction between the law article community and document nodes. This method enables a more thorough incorporation of information from law articles. The approach involves increasing the computation of the enhanced attention mechanism for both law article community and document nodes. Through our model, we are able to fully incorporate knowledge from law articles in the SE-GGNN each time the graph information is updated.

$$
\begin{aligned}
H_l^t &= [\boldsymbol{h}_{fd}^t; \mathbf{h}_c^t] \\
\hat{\boldsymbol{H}}_l^t &= \text{Self Attention}(\boldsymbol{H}_l^t) \\
\hat{\boldsymbol{H}}_l^t &= [\hat{\boldsymbol{h}}_{fd}^t; \hat{\boldsymbol{h}}_c^t]
\end{aligned}
\tag{19}
$$

where $\boldsymbol{H}_l$ is the fact node vector and the law article vector, $\boldsymbol{h}_{fd}$ is the vector representation of fact description nodes, $\boldsymbol{h}_c$ is the node vector representation of the corresponding law article community, $\hat{\boldsymbol{h}}_{fd}, \hat{\boldsymbol{h}}_c$ is the updated node features of document and law article community and we use them to update nodes set. $Self\ Attention$ is self-attention calculation.

4.3. Inter-graph fusion mechanism

After completing the intra-graph information interaction, the nodes within each graph need to update their respective intra-graph representations. The next step is to perform inter-graph fusion to update all node states. The paper constructs five graphs where nodes contain semantic units with different intra-graph information. To encode both contextual within the same graph and inter-graph semantic information, we use the state update process that models the nodes' state update process separately using different parameters. Multiple text graphs have been found to be effective for fusion (Dai et al., 2022; Liu et al., 2020). Drawing inspiration from the soft attention approach (Zhang et al., 2021), we propose a mechanism for inter-graph information fusion using multiple graphs (MG-Fusion). The inter-graph fusion process is divided into three distinct parts: (1) local contextual information group fusion, (2) semantic information group fusion, and (3) fusion of the two groups. In particular, we select three key features by experimentation: the local contextual group, represented by the co-occurrence graph; the semantic group, represented by the cosine similarity graph; and the two group, represented by the local contextual group. We use these selected features for subsequent computations.

To begin, we conduct the multi-graph fusion operation within the local contextual information group to obtain its representation. This is achieved through the use of the co-occurrence graph as the key feature, with the application of the following calculation formula:

$$\begin{aligned} \alpha_{i,j}^{a} &= \sigma(W_{a}^{oc(t)}\boldsymbol{h}_{i}^{oc(t)} + W_{a}^{pmi(t)}\boldsymbol{h}_{j}^{pmi(t)} + b_{a}) \\ \boldsymbol{h}_{i}^{a(t)} &= \boldsymbol{h}_{i}^{oc(t)} + \sum_{j\in N(i)} \alpha_{i,j}^{a} \otimes \boldsymbol{h}_{j}^{pmi(t)} \end{aligned} \tag{20}$$

where $\boldsymbol{h}_{i}^{oc(t)}$ is the node i hidden vector of the co-occurrence graph at t layer and $\boldsymbol{h}_{j}^{pmi(t)}$ is the j node hidden vector of the PMI graph at t layer. The fusion weight $\alpha_{i,j}^{a}$ is calculated using a sigmoid function. N refers to the set of neighboring graph nodes i, while W_{a}^{oc} and W_{a}^{pmi} are both trainable parameters. The fused feature vector is obtained through the use of the ⊗ operation and element-wise summation. Furthermore, $\boldsymbol{h}_{i}^{a(t)}$ is the local contextual group feature of node i. The second component of fusion pertains to semantic information, for which we opt to employ a cosine similarity graph as the key feature by experiment.

$$\begin{aligned} \alpha_{i,j}^{b} &= \sigma(W_{b1}^{cos(t)}\boldsymbol{h}_{i}^{cos(t)} + W_{b1}^{amr(t)}\boldsymbol{h}_{j}^{amr(t)} + W_{b1}^{sd(t)}\boldsymbol{h}_{j}^{sd(t)} + b_{b1}) \\ \beta_{i,j}^{b} &= \sigma(W_{b2}^{cos(t)}\boldsymbol{h}_{i}^{cos(t)} + W_{b2}^{amr(t)}\boldsymbol{h}_{j}^{amr(t)} + W_{b2}^{sd(t)}\boldsymbol{h}_{j}^{sd(t)} + b_{b2}) \\ \boldsymbol{h}_{i}^{b(t)} &= \boldsymbol{h}_{i}^{cos(t)} + \sum_{j\in N(i)} \alpha_{i,j}^{b} \otimes \boldsymbol{h}_{j}^{amr(t)} + \sum_{j\in N(i)} \beta_{i,j}^{b} \otimes \boldsymbol{h}_{j}^{sd(t)} \end{aligned} \tag{21}$$

where $\boldsymbol{h}_{i}^{cos(t)}$ is the node i hidden vector of the cosine similarity graph at t layer, $\boldsymbol{h}_{j}^{amr(t)}$ is the j node hidden vector of the AMR graph and $\boldsymbol{h}_{j}^{sd(t)}$ is the j node hidden vector of semantic dependency graph. $\boldsymbol{h}_{i}^{b(t)}$ is the semantic group feature of the node i.

Using the calculations outlined above, our method produces two distinct sets of features that represent their respective groups. The final step of our fusion method involves combining these two sets of representations using the following formula.

$$\begin{aligned} \alpha_{i,j} &= \sigma(W^{t}\boldsymbol{h}_{i}^{a(t)} + W^{t}\boldsymbol{h}_{j}^{b(t)} + b) \\ \boldsymbol{h}_{i}^{t} &= \boldsymbol{h}_{i}^{a(t)} + \sum_{j\in N(i)} \alpha_{i,j} \otimes \boldsymbol{h}_{j}^{b(t)} \end{aligned} \tag{22}$$

where $\boldsymbol{h}_{i}^{t}$ is the node i hidden vector after multi-graph fusion at t layer. Through our calculations, we demonstrate the full merging of node features from various types of graphs. One advantage of our multi-graph fusion approach is the ability to dynamically determine the degree of fusion for each graph type. To better understand this mechanism, we decompose it and present it graphically in Fig. 5. We also explored the fusion of node vectors using direct concatenation or summing of corresponding elements. Unfortunately, these methods do not yield satisfactory results, as shown in the experimental section. The main issue is that these simple approaches cause a loss of valuable information that is crucial for the success of the LJP task.

4.4. Readout

After undergoing t iterations of intra-graph and inter-graph information passing, we can obtain the node feature vector. Next, we remove both the law article community nodes and document nodes, resulting in the new $\boldsymbol{H}$. To make full graph inferences, the readout process is a standard method used by graph neural networks. In our approach, we introduce a new readout method which involves a self-attention calculation in the dimension of t iterations. This allows us to obtain the updated feature representation of each node. The calculation formula is as follows.

$$\begin{aligned} \boldsymbol{H}_{i} &= \{\boldsymbol{h}_{i}^{1}, \dots, \boldsymbol{h}_{i}^{t}\} \\ \boldsymbol{H}_{i}^{F} &= \mathrm{Self\ Attention}(\boldsymbol{h}_{i}^{1}, \dots, \boldsymbol{h}_{i}^{t}) \\ \boldsymbol{h}_{i}^{F} &= \boldsymbol{h}_{i}^{1} \end{aligned} \tag{23}$$

where, $\boldsymbol{H}_{i}$ is the set of graph node i vectors from 1th to tth iteration, Self Attention is the normal self-attention calculation, $\boldsymbol{h}_{i}^{F}$ is the node i feature vector after the calculation through attention mechanism. Afterwards, we can get $\boldsymbol{h}_{F} = \{\boldsymbol{h}_{1}^{F}, \boldsymbol{h}_{2}^{F}, \dots, \boldsymbol{h}_{n}^{F}\}$.

To proceed, we employ mean-pooling and max-pooling techniques to obtain the feature vectors for the entire graph by utilizing the following equations:

$$\boldsymbol{h}_{G} = \mathrm{MeP}(\boldsymbol{h}_{F}) + \mathrm{MaP}(\boldsymbol{h}_{F}) \tag{24}$$

where $\boldsymbol{h}_{F}$ is the nodes vector set from node 1 to n, $MeP(-)$ is Mean-Pooling operation and $MaP(-)$ is Max-Pooling operation, $\boldsymbol{h}_{G}$ is the vector representation of the whole graph after our readout.

Table 2
Dataset details.

Dataset	CAIL-small	CAIL-big
Training set cases	101 690	1 588 768
Test set cases	20 338	185 212
Law articles	103	118
Charges	119	130
Terms of penalty	11	11

4.5. Multi-task learning

LJP involves multiple learning tasks. At the prediction layer, we introduce a softmax classifier which takes the vector $\mathbf{h}_G$ as input, using the following equation.

$$\hat{y}_i = softmax(W_g^i \boldsymbol{h}_G + b_g^i) \tag{25}$$

where i represents the total number of tasks. Our training objective is to minimize the cross-entropy loss by comparing the ground truth label to the predicted label for each sub-task. We arrive at the overall prediction loss by summing the losses of all sub-tasks together.

$$\mathcal{L} = -\sum_{i=1}^{3} \sum_{k=1}^{N_i} y_{i,k} \log(\hat{y}_{i,k}) \tag{26}$$

where N_i represents the number of distinct labels for task i. Additionally, $y_{i,k}$ refers to the one-hot vector corresponding to the ground truth label of task i, while $\hat{y}_{i,k}$ denotes the predicted label. In addition, the selection of the law article community nodes is also included in the calculation of the loss function, which also uses cross-entropy loss.

5. Experiments

5.1. Datasets

To demonstrate the efficacy of our approach, we have employed the CAIL-small and CAIL-big datasets, which are part of the Chinese AI and Law challenge (Xiao, Zhong, Guo, Tu, Liu, Sun, Feng, Han, Hu, Wang, & Xu, 2018). These datasets consist of cases that comprise a fact description, the matching law article, the charge, and the subsequent penalty term. Prior to commencing our analysis, we curate the datasets by eliminating instances that were less than 10 words long and consisted of only a single charge and law article. Based on the guidance from Zhong et al. (2018), we then proceed to select charges that had a minimal count of 100 cases. Table 2 reveals some important statistics about the CAIL datasets. In the CAIL-small dataset, we find that the Training set has a total of 101,690 samples and the Test set has 20,338 samples. This dataset includes 103 different law articles, 119 charges, and 11 terms. Meanwhile, the CAIL-big dataset has a larger Training set with 1,588,768 samples and a Test set of 185,212 samples. This dataset includes 118 law articles, 130 charges, and 11 terms.

5.2. Baselines

To conduct comparative experiments, we have carefully chosen nine baseline models that demonstrate exceptional performance.

- **TFIDF+SVM** (Suykens & Vandewalle, 1999). This method uses TF-IDF is adopted to construct feature information for text and SVM as the classifier.
- **CNN** (Kim, 2014). A CNN with multiple filters is utilized to extract the feature information of the text and finally uses softmax as the classifier.
- **RCNN** (Lai et al., 2015). The method flexibly combines RNN and CNN to construct new models to improve text classification performance.
- **HARNN** (Yang et al., 2016). This method uses RNN with a hierarchical attention mechanism as text feature extractor and softmax as the classifier.
- **FLA** (Luo et al., 2017). The text feature extractor exploits the attention mechanism to incorporate the case corresponding legal information for the prediction effect.
- **TOPJUDGE** (Zhong et al., 2018). This approach is a multi-task framework that captures the topological dependencies between LJP subtasks through a directed acyclic graph structure.
- **Text-GCN** (Yao et al., 2019). This method innovatively proposes a graph convolutional network for text classification with good results.
- **MPBFN-WCA** (Yang et al., 2019). This method uses a multi-task learning framework for LJP with multi-perspective forward prediction and backward verification.
- **LADAN** (Xu et al., 2020b). This work proposes an end-to-end LJP framework that is constructed with graph neural networks to solve the problem of confusion-prone law articles.

Table 3
Legal judgment prediction results on CAIL-small.

	Law articles				Charges				Terms of penalty			
	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
TFIDF + SVM	76.52	43.21	40.12	39.68	79.81	45.86	42.72	42.77	33.32	27.66	24.99	24.64
FLA	77.72	75.21	74.12	72.78	80.98	79.11	77.92	76.77	36.32	30.81	28.22	27.83
CNN	78.61	75.86	74.6	73.59	82.23	81.57	79.73	78.82	35.2	32.96	29.09	29.68
RCNN	79.12	76.58	75.13	74.15	82.50	81.89	79.72	79.05	35.52	33.76	30.41	30.27
HARNN	79.73	75.05	76.54	74.67	83.41	82.23	82.27	80.79	35.95	34.5	31.04	31.18
TOPJUDGE	79.79	79.52	73.39	73.33	82.03	83.14	79.33	79.03	36.05	34.54	32.49	29.19
Text-GCN	79.81	79.65	73.42	73.37	82.33	83.19	79.20	78.97	35.97	34.66	32.54	29.23
MPBFN	79.12	76.30	76.02	74.78	82.14	82.28	80.72	80.72	36.02	31.94	28.60	29.85
LADAN	82.34	78.79	77.59	76.80	84.83	83.33	82.80	82.85	39.35	36.94	33.25	34.05
LA-MGFM	**84.95**	**83.91**	**83.32**	**82.93**	**89.65**	**88.74**	**88.90**	**88.96**	**43.01**	**41.94**	**40.06**	**41.04**

Table 4
Legal judgment prediction results on CAIL-big.

	Law articles				Charges				Terms of penalty			
	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
TFIDF + SVM	89.93	68.56	60.58	61.25	85.81	69.76	61.92	63.51	54.13	39.15	37.62	39.14
FLA	93.22	72.81	64.27	66.57	92.48	76.21	68.12	69.97	57.66	43.01	38.89	41.63
CNN	95.79	82.79	75.15	76.62	95.23	86.57	78.93	81.02	55.41	45.23	38.73	39.96
RCNN	95.98	82.93	75.26	77.13	95.50	87.89	79.03	81.65	55.62	45.43	38.88	40.17
HARNN	96.01	82.99	75.58	77.38	95.62	87.93	79.27	81.79	56.11	44.21	40.57	41.87
TOPJUDGE	95.81	84.41	74.36	76.67	95.73	87.99	79.49	81.93	57.29	47.35	42.61	44.03
Text-GCN	95.69	84.24	74.22	76.58	95.60	87.89	79.28	81.82	57.2	47.17	42.53	43.91
MPBFN	96.06	85.25	74.82	78.36	95.98	89.16	79.73	83.20	58.14	45.86	39.07	41.3
LADAN	96.57	86.22	80.78	82.36	96.45	88.51	83.73	85.35	59.66	51.78	45.34	46.93
LA-MGFM	**97.98**	**88.97**	**87.21**	**87.95**	**97.59**	**91.74**	**89.12**	**90.13**	**63.05**	**54.29**	**52.68**	**53.56**

5.3. Experimental settings

We have implemented our model using PyTorch. For optimization, we have employed the Adam algorithm (Kingma & Ba, 2015), with a learning rate of 0.01 and dropout set to 0.5 (SE-GGNN). Since our work involves Chinese, we have used the THULAC word segmentation tool (https://github.com/thunlp/THULAC-Python), which has proven to be highly effective. We have also used mini-batch training to avoid overloading memory, and have set a threshold of 5 for discarding infrequent words while constructing the vocabulary. We utilize LawFormer as our word embedding to obtain embedding-based distances. In order to ensure efficient training, if the validation loss fails to decrease for 100 consecutive epochs, we halt the training process. Additionally, we conduct parameter searches to determine optimal values for the dropout rate, examining values of 0.3, 0.5, and 0.7, as well as the batch size, exploring choices of 8, 16, 32, and 64. We have utilized Deep Graph Library (DGL: https://docs.dgl.ai/index.html) for graph reasoning and scikit-learn (https://scikit-learn.org/stable/) to quantify the metrics in our code implementation. In our law article feature extractor, we utilize three iterations to differentiate between the various law article communities and obtain their respective representations. With a hidden vector dimension of 200 and 3 interaction steps, our SE-GGNN was successfully executed. Furthermore, our reported results are the average values derived from three separate runs with different random initializations.

Baseline Evaluation: For the baseline models in the comparison experiments, we reproduce results of CNN, RCNN, HARNN and GCN based on the open-source code and other results are the same as the results in their original papers (Luo et al., 2017; Xu et al., 2020b; Yang et al., 2019; Zhong et al., 2018). We use 300 dimensional GloVe (Pennington et al., 2014) word embeddings for the baseline models using pre-trained word embeddings. For CNN based model in baselines, we set the maximum document length to 512 words, the number of filters is 256, and the length of sliding window is $\{2, 3, 4, 5\}$ respectively as Kim (2014). For RNN-based model in baselines, we set the maximum sentence length to 64 words and maximum document length to 64 sentences. At the same time, we use the porch framework to construct neural networks. In the training part, we set the learning rate of Adam optimizer as 0.001, and the dropout probability as 0.5. Both the CNN based models and the RNN based models have batch sizes of 128. We train every model for 16 epochs and evaluate the final model on the test set. For Text-GCN, the number of interaction steps is 3 and the sliding window size is 3. For the first convolution layer, we set the embedding dimension to 200. We tuned other parameters and set the learning rate as 0.02, dropout rate as 0.5. Following the setting of Yao et al. (2019), we set window-size 20 to obtain the co-occurrence information. We train Text-GCN for a maximum of 200 epochs using Adam and stop training if the validation loss does not decrease for 10 consecutive epochs.

5.4. Experimental results

This paper considers accuracy, precision, recall, and macro F1 values as comparison metrics. Tables 3 and 4 demonstrate that our method surpasses the baseline model in all metrics for both CAIL-small and CAIL-big. Regarding the LJP subtasks (charge

Table 5
Ablation experiments results on CAIL-small.

	Law articles				Charges				Terms of penalty			
	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
LA-MGFM	**84.95**	**83.91**	**83.32**	**82.93**	**89.65**	**88.74**	**88.90**	**88.96**	**43.01**	**41.94**	**40.06**	**41.04**
LA-MGFM$_{only\ A-Group}$	82.34	81.81	81.95	80.89	86.65	84.32	84.39	84.01	41.71	40.52	40.01	39.90
LA-MGFM$_{only\ B-Group}$	80.17	79.09	79.52	78.95	83.88	82.69	82.22	81.90	38.87	37.95	37.09	36.97
LA-MGFM$_{B-Group+Coo}$	82.37	81.46	81.82	80.74	86.88	84.37	85.70	84.22	42.17	40.21	40.31	40.04
LA-MGFM$_{B-Group+PMI}$	82.33	81.39	81.67	80.52	86.49	83.62	83.59	82.82	41.62	39.47	38.99	39.02
LA-MGFM$_{A-Group+Cos}$	83.10	82.87	82.39	81.64	87.18	86.20	86.12	86.11	42.25	41.57	40.87	40.61
LA-MGFM$_{A-Group+AMR}$	81.97	81.03	81.43	80.24	85.75	82.67	82.74	81.81	40.72	39.08	38.91	38.37
LA-MGFM$_{A-Group+SD}$	81.82	80.95	81.41	80.42	84.35	82.51	81.96	80.99	40.21	38.86	38.31	38.08
LA-MGFM$_{use\ GF1}$	78.65	77.33	76.12	76.32	81.35	80.07	78.08	78.34	36.99	34.51	34.11	34.12
LA-MGFM$_{use\ GF2}$	78.25	76.97	75.52	75.98	80.95	79.41	78.15	78.24	36.52	34.11	33.85	33.77
LA-MGFM$_{w/o\ LAC\ Node}$	79.02	77.91	76.35	76.81	81.69	80.17	78.95	78.91	37.01	34.66	34.32	34.42
LA-MGFM$_{Key_A:PMI}$	83.21	83.03	82.71	81.27	88.33	87.12	87.04	87.62	42.10	42.96	39.91	40.21
LA-MGFM$_{Key_B:AMR}$	82.72	82.87	82.03	80.92	87.89	86.89	86.81	86.85	41.75	42.13	38.99	39.64
LA-MGFM$_{Key_B:SD}$	82.51	82.71	81.87	80.46	87.24	86.61	86.54	86.36	41.29	41.91	38.57	39.28
LA-MGFM$_{Key_Group:B}$	83.14	83.06	82.59	81.14	88.35	87.01	86.97	87.48	41.94	42.82	39.88	40.14
LA-MGFM$_{lawa}$	84.38	83.14	82.91	82.09	89.51	88.2	88.24	88.71	42.67	**41.96**	**40.12**	**41.07**

prediction, law article prediction, term of penalty prediction), our model attains an improvement of 1.41–2.61, 1.14–4.82, and 3.39–3.66 percentage points for accuracy, precision gains of 2.75–4.26, 3.23–5.41, and 2.51–5 percentage points, respectively, and recall improvements of 5.73–6.43, 5.39–6.1, and 6.81–7.34 percentage points. Furthermore, our method achieves a higher F1 value with 5.59–6.13, 4.78–6.11, and 6.63–6.99 percentage points compared to the previous SOTA LADAN in both datasets. There are three points that can be attributed to the reasons for this: 1. This paper outlines five approaches to constructing text graphs that can extract rich text features. Additionally, we present two kinds of local context-level graphs and three kinds of semantic-level graphs, which greatly enhance the data input and demonstrate a multimodal approach. 2. We utilize GGNN for the graph encoding layer to alleviate the over-smoothing effects of the traditional GCN. To further enhance GGNN's performance, we augment it with sememe information and related law article community knowledge. This integration facilitates more precise word meaning identification by the graph nodes, and also enables the fusion of related law articles during each information passing and updating process. 3. The multi-graph fusion mechanism is a crucial element in our method. To better understand its function, we separate text graphs into two distinct groups, and use this mechanism to thoroughly combine the local contextual and semantic information in each.

In addition, it is evident from Table 4 that all models demonstrate better performance on CAIL-big in comparison to CAIL-small. This highlights the significance of adequate data volume in the training of models. Furthermore, the CNN and RNN-based methods outperform traditional machine learning techniques such as TFIDF+SVM. Between the CNN and RNN models, the RNN-based model achieves superiority due to its capacity to model language more effectively, thus generating a more effective representation of textual features. As the dataset has unevenly distributed data, it is essential to compare the F1 values. In CAIL-big, LA-MGFM outperforms LADAN (+5.59, +4.78, +6.63), MPBFN-WCA (+9.59, +6.93, +12.26), and TOPJUDGE (+11.28, +8.2, +9.53) in the three subtasks based on F1 values, indicating that LA-MGFM attains significant outcomes in fusing the information of law articles.

5.5. Ablation experiments

In Table 5, we validate the effectiveness of each component of our model incrementally by performing ablation experiments on CAIL-small. We begin by reporting our model without constructing a local contextual information group or a semantic information group. LA-MGFM$_{only\ A-Group}$ refers to only building graphs of local contextual information group. LA-MGFM$_{only\ B-Group}$ refers to only building graphs of the semantic information group. It can be seen that the performance of the model falls when either group is used on its own, which points to an irreplaceable role for both groups in the feature extraction of textual information. In addition, performance drops more with the use of the semantic group alone, indicating that the local contextual group contributes more than the semantic group.

In Table 5, we have tested the validity of each component of our model through ablation experiments. We begin by reporting our model without constructing either the local contextual information group or the semantic information group. Next, we have analyzed the validity of each group independently. $textLA-MGFM_{onlyA-group}$ represents the model with only the local contextual information group constructed, while $textLA-MGFM_{onlyB-group}$ represents the model with only the semantic information group constructed. It is clear that the model's performance declines when only one group is used, highlighting the critical role both groups play in extracting features from textual information. Additionally, the local contextual group appears to have a bigger impact on performance than the semantic group, since the model's performance worsens more when only the semantic group is utilized.

To verify the contribution of each text graph in the model, we choose either the local contextual or semantic group and keep it fixed, using only one text graph from the non-fixed group. When the semantic information group is fixed, we use co-occurrence graphs from the local contextual group and denote it as $textLA-MGFM_{B-Group+Coo}$. Alternatively, when the semantic

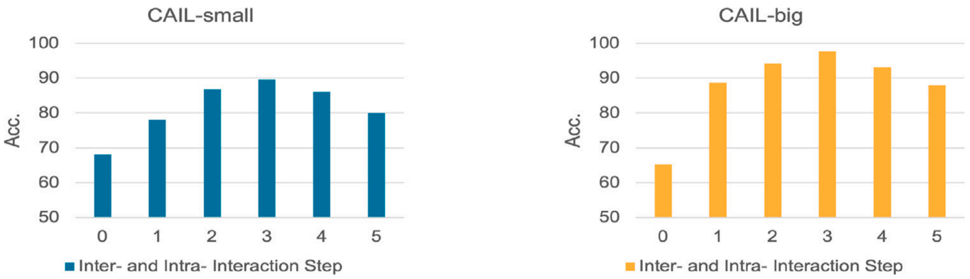


Fig. 6. Effect of different times inter- and intra-graph interaction.

information group is fixed, we use the PMI graph from the local contextual group and denote it as $textLA - MGFM_{B-Group+PMI}$. $textLA - MGFM_{A-Group+Cos}$ refers to fixing the local contextual information group while using the cosine similarity graph in the semantic information group. $textLA - MGFM_{A-Group+AMR}$ also fixes the local contextual information group but uses the AMR graph in the semantic information group. Finally, $textLA - MGFM_{A-Group+SD}$, fixes the local contextual information group and uses the SD graph in the semantic information group. When examining Table 5, it becomes evident that each of the five graphs serves a unique purpose, thereby demonstrating the soundness of our constructions. Moreover, the outcomes of $textLA - MGFM_{B-Group+Coo}$ and $textLA - MGFM_{B-Group+PMI}$ bear minimal variation, suggesting that co-occurrence and PMI graphs possess comparable functions. When we fixing the local content information groups, we have discovered that using cosine similarity graph in semantic information group led to the most favorable outcomes. This finding indicates that the complicated graph construction techniques, such as AMR and SD, may not be essential for the LJP task, despite providing higher levels of semantic information.

We then conduct experiments on various multi-graph fusion methods. Initially, we utilize the vector concatenation method (LA-MGFM$_{useGF1}$) and subsequently tested the vector element addition method (LA-MGFM$_{useGF2}$). However, the outcomes of both methods are significantly weaker in comparison to our multi-graph fusion mechanism. This suggests that these methods result in a loss of valuable information and negatively impact the model's performance. In contrast, our approach employs a flexible and dynamic fusion technique, which enables us to extract more effective features.

We conduct additional experiments to confirm the significance of the law article community node. We refer to the model without this node as LA-MGFM$_{w/o\ LAC\ Node}$. Our findings show that the performance metric decreases when the law article community node is removed from the model, with a potential decrease of up to 4 percentage points. This observation further highlights the effectiveness of the law article knowledge fusion approach proposed in this paper.

Otherwise, we validate the key graph (group) selection for inter-graph information interaction. LA-MGFM$_{Key_A:PMI}$ indicates that PMI graph is used as the key graph in the local contextual information group. LA-MGFM$_{Key_B:AMR}$ and LA-MGFM$_{Key_B:SD}$ indicates that AMR or SD graph is used as the key graph in the semantic information group, respectively. LA-MGFM$_{Key_Group:B}$ indicates that the semantic information group is used as the key group. Based on our findings, it is evident that any alterations made to the selection of key graphs or groups result in a decline in the model's performance. This further validates our decision to prioritize co-occurrence graphs, cosine similarity graphs, and local contextual information group as the key components of our model.

Finally, we investigate the impact of utilizing another word segmentation tool on the model. The LA-MGFM model, specifically, employed THULAC for segmentation, which we augment with extensive judicial domain dictionaries. We utilize the lawa (https://github.com/ShenDezhou/lawa) as compared tool, specifically developed for judicial domain texts, without the need for an external dictionary. Our results show that the model using the lawa segmentation tool outperforms THULAC in certain metrics for term of penalty prediction. However, it slightly underperforms compared to the model using THULAC in charge and law article prediction. Based on the aforementioned observations, it can be inferred that the performance of models utilizing similar word segmentation tools is relatively similar.

5.6. Parameter sensitivity

We design three parameter sensitivity experiments using our model to assess its robustness in terms of hyper-parameter selection. For this experiment, we utilize the CAIL-small and CAIL-big for performance testing in charge prediction. Firstly, we test the impact of various interaction steps on the accuracy of charge prediction. Secondly, we will examine the effect of different window sizes on the accuracy of constructing a co-occurrence graph. Lastly, we validate the effect of different similarity thresholds on the accuracy of constructing the cosine similarity graph.

5.6.1. Analysis of different times of steps inter- and intra-graph interaction

To test the performance of our SE-GGNN and Multi-graph fusion, we can implement our approach with multiple digits, as the number of interaction steps has a significant impact. In this section, we analyze the performance of different numbers of interaction steps to determine the optimal value. Specifically, we compare the performance of interaction steps ranging from 0 to 5. Based on Fig. 6, it is evident that the optimal performance is achieved with 3 interaction steps, outperforming all other numbers. Furthermore, the results for number 2 are also satisfactory, suggesting that 2 interactions may be a viable option if computational resources are limited. Overall, these findings validate our decision to use 3 interaction steps as the optimal choice.

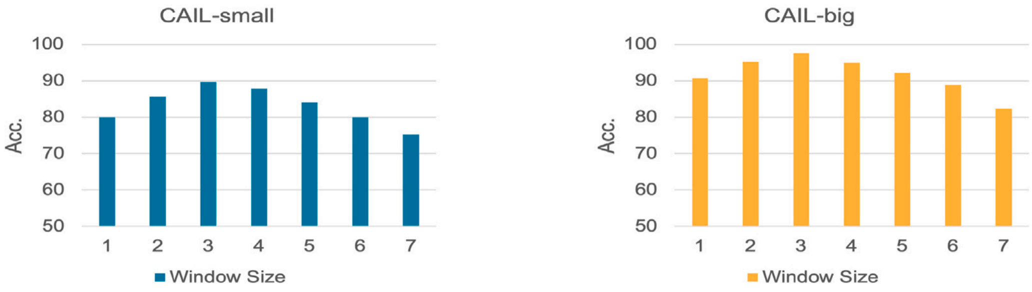


Fig. 7. Effect of different window sizes of co-occurrence graph.

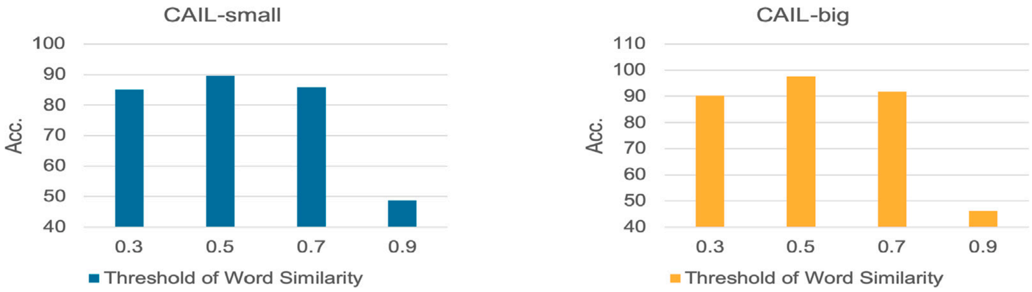


Fig. 8. Effect of threshold of cosine similarity graph.

5.6.2. Analysis of different window sizes of co-occurrence graph

In order to evaluate the performance of various window sizes for the co-occurrence graph, we conduct an experiment where we construct co-occurrence graphs using different window sizes. To demonstrate the efficacy of our chosen window size, we also implement co-occurrence graphs using fixed window sizes ranging from 1 to 7. From Fig. 7, we can infer that a window size of 7 yields the poorest performance among all the window sizes. This is not surprising as a window size of 7 fails to capture the diverse relatedness between nodes. Additionally, a window size of 1 results in relatively low performance, suggesting that it can only capture feature information for a very limited region. Conversely, a window size of 3 produces the best results on the metric. The reason for this is that a window size of 3 aligns with the typical approach to sequence modeling, allowing for adequate co-occurrence information to be gathered.

5.6.3. Analysis of different threshold of cosine similarity graph

To analyze the impact of threshold on the cosine similarity graph, we utilize various threshold numbers, including $\{0.3, 0.5, 0.7, 0.9\}$, and compare their results, as depicted in Fig. 8. Our findings indicate that the 0.5 threshold yielded the most favorable outcome. This is due to the fact that a 0.5 similarity threshold allows text graphs to include word pairs with some degree of relevance while effectively eliminating word pairs that lack relevance, even if they possess some similarity. Text graphs can retain a significant amount of semantic information. Fig. 8 illustrates that the trend declines quickly when the threshold is above or below 0.5. This suggests that selecting alternative thresholds for constructing the text similarity graph is not an effective way to model semantics.

5.7. Other experiments

To verify the impact of the pre-trained model, we conduct an experiment using cail-small for the charge prediction task. We selected three pre-training models - BERT, LABERT, and LawFormer - and use them solely to predict the charges. We attached a layer of classification headers to the pre-trained models and fine-tune them to derive the results. Then we use Skip-gram, BERT, and Legal ELECTRA to replace Lawformer on word representation used in our method. In Fig. 9, it is evident that BERT's performance is the weakest when solely relying on the pre-trained model for charge prediction. This implies that models trained on general domain datasets may not be suitable for specific domains. While the Lawformer model shows the best performance, it still lags behind the LA-MGFM model presented in this paper. The model using lawformer as word representation shows the best performance, while the model using skip-gram has the worst performance. However, LA-MGFM using skip-gram still performs better than the LADAN model, which also uses skip-gram as word vectors. This suggests that the superiority of LA-MGFM is not solely due to its use of pre-trained models in the legal domain.

We conduct experiments to confirm our model's ability to handle few-shot data. To do this, we filter the CAIL-small dataset based on the number of cases, resulting in three categories: 10–50, 50–100, and over 100. We then compare our model to TOPJUDGE,

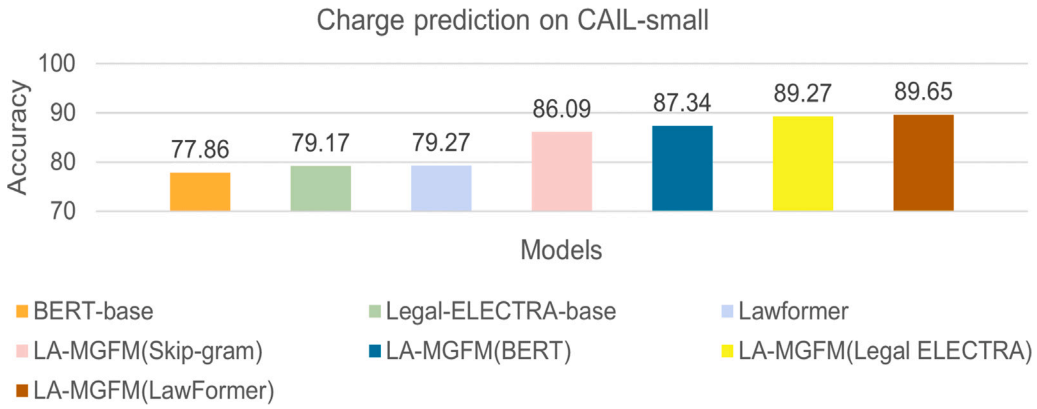


Fig. 9. Experimental results using different pre-trained models as backbone networks or word representations.

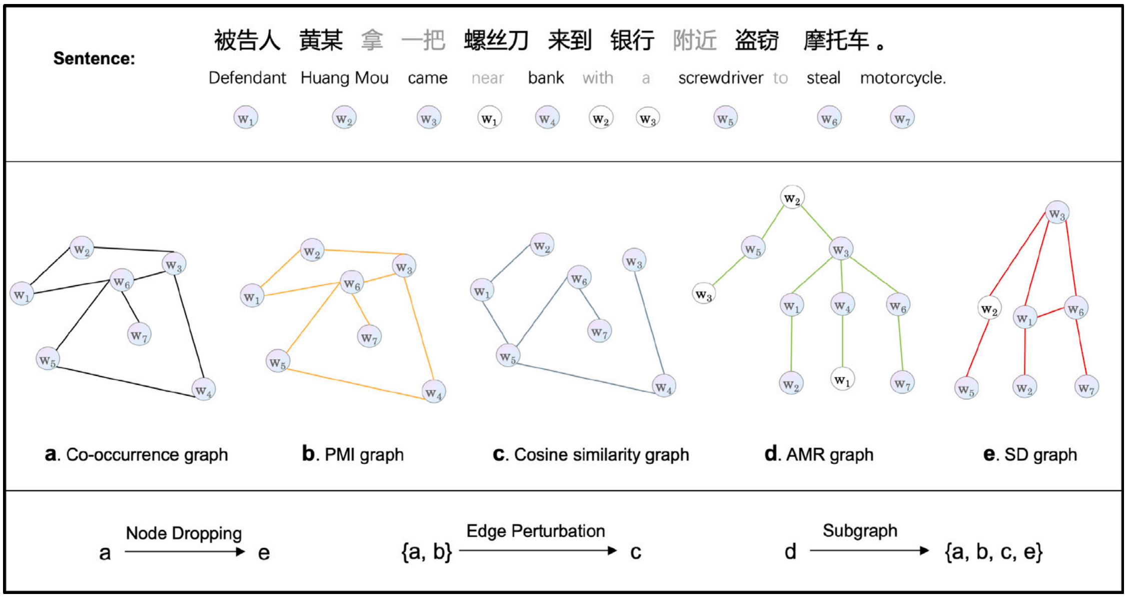

Sentence:
被告人 黄某 拿 一把 螺丝刀 来到 银行 附近 盗窃 摩托车。
Defendant Huang Mou came near bank with a screwdriver to steal motorcycle.
a. Co-occurrence graph
b. PMI graph
c. Cosine similarity graph
d. AMR graph
e. SD graph
a Node Dropping e
{a, b} Edge Perturbation c
d Subgraph {a, b, c, e}

Fig. 10. A sentence example of five kinds of text graphs constructed in this paper.

Table 6
Performance of the model in few-shot charge prediction.

Number	10 < n ≤ 50	50 < n ≤ 100	n > 100
TOPJUDGE	36.58	56.14	82.03
GCN	43.59	58.19	81.97
MPBFN	39.56	57.15	83.46
LADAN	42.12	59.18	86.04
LA-MGFM	**51.19**	**66.43**	**90.97**

GCN, MPBFN, and LADAN, using F1 values as the metric for predicting charges. As shown in Table 6, our model outperforms all comparison models in handling few-shot data, demonstrating excellent experimental results. The proposed method demonstrates significant advantages in handling few-shot datasets, as evidenced by the improved F1 values compared to TOPJUDGE, GCN, MPBFN, and LADAN methods. Specifically, the F1 values are 14.61, 7.6, 11.63, and 9.07 percent point higher between 10 and 50, 10.29, 8.24, 9.28, and 7.25 percent point higher between 50 and 100, and 8.94, 9, 7.51, and 4.93 percent point higher than 100, respectively.

After analyzing several cases, we have come to the conclusion that our approach to creating five types of text graphs is akin to the graph data augmentation theory. Furthermore, we have incorporated a multi-graph fusion mechanism into our method. By

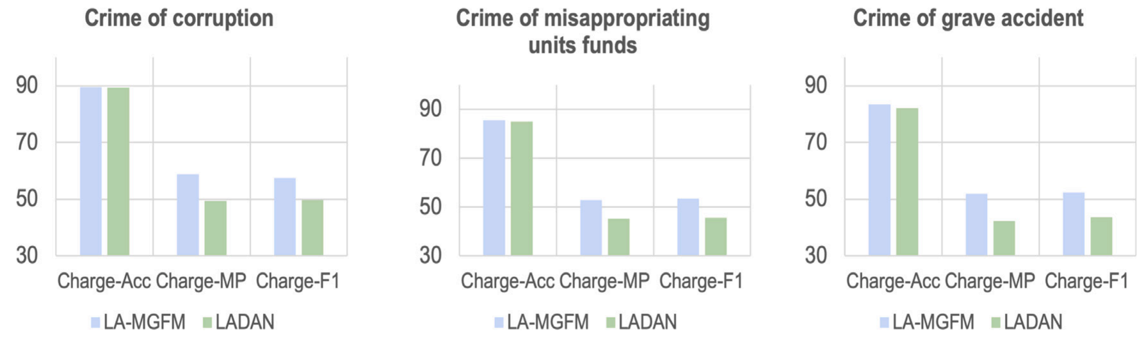


Fig. 11. Prediction performance of three confusing charges.

combining these two essential processes, our approach outperforms other baselines in few-shot scenarios. To elaborate, we have provided a theoretical foundation for each graph data augmentation method and have identified which graphs in Fig. 10 align with these theories.

- **Node Dropping**: Take a given Graph G and randomly discard the node that determines the ratio, as well as its associated connections. This approach is based on prior knowledge that discarding certain nodes does not affect the semantics of Graph G. For instance, in our example, dropping a node can result in the generation of a new graph, denoted as e, through a process of node dropping.
- **Edge Perturbation**: Randomly adding and deleting edges on the connectivity of Graph G. The prior knowledge suggests that the semantics of Graph G are somewhat resilient to changes in edge connections. For instance, in our example, the a and b graphs are capable of producing the c graph through edge perturbation.
- **Subgraph**: A subgraph of Graph G is obtained through sampling. With prior knowledge, the local structure of Graph G can be preserved while still maintaining its semantics. If edge weights are disregarded, the d graph has the ability to generate the a, b, c, and e graphs.

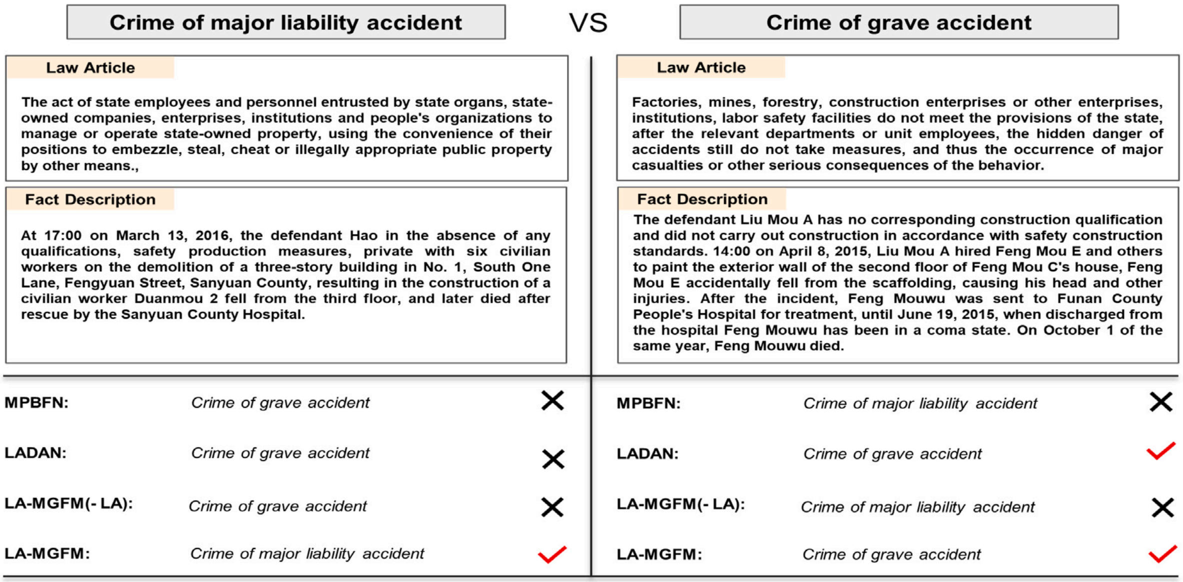
5.8. Case study of confusing charges

To demonstrate the model's ability to distinguish between confusing charges in a more intuitive manner, we have selected three charges (crime of grave accident, crime of misappropriating units funds and crime of corruption) and extracted the cases that correspond to the most confusing charges as a new test set from the CAIL-small test set. We evaluate the performance of the model in charge prediction using accuracy, precision, and F1 values as evaluation metrics. For comparison, we have chosen LADAN as the baseline model and report its experimental results. Both models have been trained based on the CAIL-small dataset. Fig. 11 shows the results. It is noted that the experimental metrics show a high accuracy but low precision, which reflects the confusion-prone nature of the charges. Additionally, while the accuracy obtained by LADAN is comparable to our model, the MP and F1 values are significantly lower. This suggests that our model is effective in terms of fusing knowledge of law articles and extracting textual features, which can improve the prediction ability of the model for confusing charges (fact descriptions) and contribute to the performance of LJP.

Furthermore, we have chosen three sets of confusing charges along with their corresponding six case descriptions (see Figs. 12–14). In regards to the models, we have chosen three comparison models: MPBFN, LADAN, and LA-MGFM. Our models have accurately predicted the results in all three examples. Out of the selected comparative models, MPBFN predicts incorrectly in all cases. LADAN predicts correctly in three of the cases due to its incorporation of knowledge of law articles. LA-MGFM (-LA) predicts incorrectly in all cases because it do not incorporate the knowledge of law articles. However, LA-MGFM (-LA) predicts correctly in all cases. This paper proposes that the LA-MGFM method can effectively address the issue of confusing charges by utilizing information from law article community. As a result, it can be demonstrated that LA-MGFM is a promising solution to this problem.

6. Discussion

In this paper, we present a highly effective method (LA-MGFM) for LJP. Our approach involves several sub-processes. First, we construct the input text into five different types of text graphs to facilitate further learning of the graph representation. Next, we utilize an intra-graph word interaction mechanism (Sememe-enhanced GGNN) and an inter-graph fusion mechanism to obtain a comprehensive legal text graph representation. Finally, we employ a readout to obtain a representation of the entire graph, which we use to generate the LJP results. In this section, we begin by exploring the connections and differences between our approach and the existing research. Following this, we outline the limitations of our study. Finally, we conclude with a summary of potential future improvements.



Crime of major liability accident VS Crime of grave accident

Law Article

The act of state employees and personnel entrusted by state organs, state-owned companies, enterprises, institutions and people's organizations to manage or operate state-owned property, using the convenience of their positions to embezzle, steal, cheat or illegally appropriate public property by other means.,

Fact Description

At 17:00 on March 13, 2016, the defendant Hao in the absence of any qualifications, safety production measures, private with six civilian workers on the demolition of a three-story building in No. 1, South One Lane, Fengyuan Street, Sanyuan County, resulting in the construction of a civilian worker Duanmou 2 fell from the third floor, and later died after rescue by the Sanyuan County Hospital.

Law Article

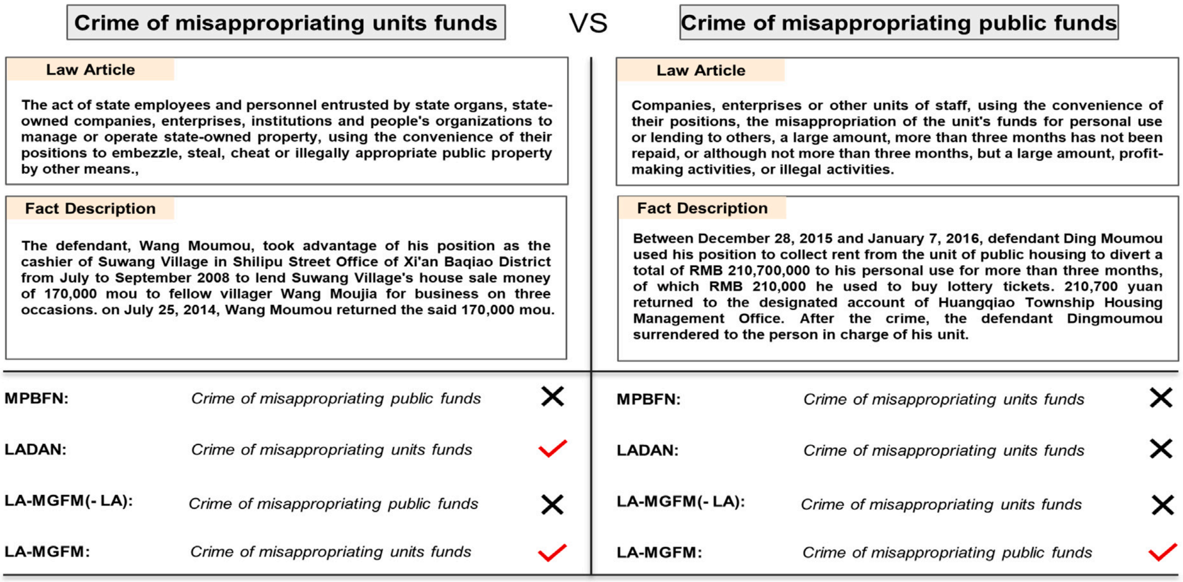
Factories, mines, forestry, construction enterprises or other enterprises, institutions, labor safety facilities do not meet the provisions of the state, after the relevant departments or unit employees, the hidden danger of accidents still do not take measures, and thus the occurrence of major casualties or other serious consequences of the behavior.

Fact Description

The defendant Liu Mou A has no corresponding construction qualification and did not carry out construction in accordance with safety construction standards. 14:00 on April 8, 2015, Liu Mou A hired Feng Mou E and others to paint the exterior wall of the second floor of Feng Mou C's house, Feng Mou E accidentally fell from the scaffolding, causing his head and other injuries. After the incident, Feng Mouwu was sent to Funan County People's Hospital for treatment, until June 19, 2015, when discharged from the hospital Feng Mouwu has been in a coma state. On October 1 of the same year, Feng Mouwu died.

MPBFN:	*Crime of grave accident*	✗	MPBFN:	*Crime of major liability accident*	✗
LADAN:	*Crime of grave accident*	✗	LADAN:	*Crime of grave accident*	✓
LA-MGFM(- LA):	*Crime of grave accident*	✗	LA-MGFM(- LA):	*Crime of major liability accident*	✗
LA-MGFM:	*Crime of major liability accident*	✓	LA-MGFM:	*Crime of grave accident*	✓

Fig. 12. Confusing charges: crime of grave accident vs. crime of major liability accident.



Crime of misappropriating units funds VS Crime of misappropriating public funds

Law Article

The act of state employees and personnel entrusted by state organs, state-owned companies, enterprises, institutions and people's organizations to manage or operate state-owned property, using the convenience of their positions to embezzle, steal, cheat or illegally appropriate public property by other means.,

Fact Description

The defendant, Wang Moumou, took advantage of his position as the cashier of Suwang Village in Shilipu Street Office of Xi'an Baqiao District from July to September 2008 to lend Suwang Village's house sale money of 170,000 mou to fellow villager Wang Moujia for business on three occasions. on July 25, 2014, Wang Moumou returned the said 170,000 mou.

Law Article

Companies, enterprises or other units of staff, using the convenience of their positions, the misappropriation of the unit's funds for personal use or lending to others, a large amount, more than three months has not been repaid, or although not more than three months, but a large amount, profit-making activities, or illegal activities.

Fact Description

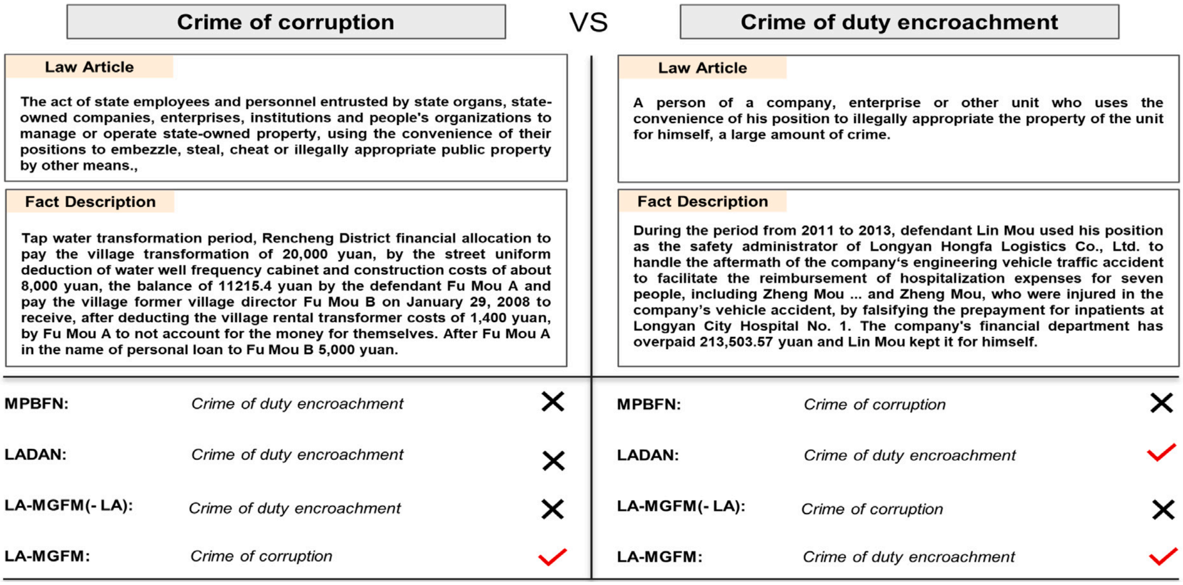
Between December 28, 2015 and January 7, 2016, defendant Ding Moumou used his position to collect rent from the unit of public housing to divert a total of RMB 210,700,000 to his personal use for more than three months, of which RMB 210,000 he used to buy lottery tickets. 210,700 yuan returned to the designated account of Huangqiao Township Housing Management Office. After the crime, the defendant Dingmoumou surrendered to the person in charge of his unit.

MPBFN:	*Crime of misappropriating public funds*	✗	MPBFN:	*Crime of misappropriating units funds*	✗
LADAN:	*Crime of misappropriating units funds*	✓	LADAN:	*Crime of misappropriating units funds*	✗
LA-MGFM(- LA):	*Crime of misappropriating public funds*	✗	LA-MGFM(- LA):	*Crime of misappropriating units funds*	✗
LA-MGFM:	*Crime of misappropriating units funds*	✓	LA-MGFM:	*Crime of misappropriating public funds*	✓

Fig. 13. Confusing charges: crime of misappropriating units funds vs. crime of misappropriating public funds.

6.1. Link and difference with existing works

Our proposed LA-MGFM can achieve good performance for LJP task. The previous LJP methods can be summarized into four types: (1) mathematical and quantitative analysis methods (Kort, 1957; Nagel, 1964), (2) methods based on traditional machine learning algorithms and feature engineering (Katz et al., 2014; Liu & Hsieh, 2006; Liu et al., 2015), (3) new models with novel network structures (Chen et al., 2019; Jiang et al., 2018; Li et al., 2019; Liu et al., 2019; Long et al., 2019; Pan et al., 2019; Wang et al., 2019), and (4) methods integrated legal knowledge into the model (Hu et al., 2018b; Kang et al., 2019; Luo et al., 2017; Wei & Lin, 2019; Xu et al., 2020b; Zhong et al., 2018). Our method is a combination of 3 and 4 methods. We design new network structures like Sememe-enhanced GGNN and multi-graph fusion mechanism. The Sememe-enhanced GGNN can mitigate the word sense confusion problem, and multi-graph fusion mechanism can efficiently combine the feature information contained in different types of text graph. Section 5 presents our experimental results that validate the intuition above.



Crime of corruption VS Crime of duty encroachment

Law Article

The act of state employees and personnel entrusted by state organs, state-owned companies, enterprises, institutions and people's organizations to manage or operate state-owned property, using the convenience of their positions to embezzle, steal, cheat or illegally appropriate public property by other means.,

Fact Description

Tap water transformation period, Rencheng District financial allocation to pay the village transformation of 20,000 yuan, by the street uniform deduction of water well frequency cabinet and construction costs of about 8,000 yuan, the balance of 11215.4 yuan by the defendant Fu Mou A and pay the village former village director Fu Mou B on January 29, 2008 to receive, after deducting the village rental transformer costs of 1,400 yuan, by Fu Mou A to not account for the money for themselves. After Fu Mou A in the name of personal loan to Fu Mou B 5,000 yuan.

Law Article

A person of a company, enterprise or other unit who uses the convenience of his position to illegally appropriate the property of the unit for himself, a large amount of crime.

Fact Description

During the period from 2011 to 2013, defendant Lin Mou used his position as the safety administrator of Longyan Hongfa Logistics Co., Ltd. to handle the aftermath of the company's engineering vehicle traffic accident to facilitate the reimbursement of hospitalization expenses for seven people, including Zheng Mou ... and Zheng Mou, who were injured in the company's vehicle accident, by falsifying the prepayment for inpatients at Longyan City Hospital No. 1. The company's financial department has overpaid 213,503.57 yuan and Lin Mou kept it for himself.

MPBFN:	Crime of duty encroachment	✗	MPBFN:	Crime of corruption	✗
LADAN:	Crime of duty encroachment	✗	LADAN:	Crime of duty encroachment	✓
LA-MGFM(- LA):	Crime of duty encroachment	✗	LA-MGFM(- LA):	Crime of corruption	✗
LA-MGFM:	Crime of corruption	✓	LA-MGFM:	Crime of duty encroachment	✓

Fig. 14. Confusing charges: crime of corruption vs. crime of duty encroachment.

Our method is based on incorporating law articles. It uses the same extra knowledge like LADAN (Xu et al., 2020b), previous SOTA baseline. LA-MGFM differs from LADAN in the following ways: 1. differences in the input layer: We use both fact descriptions and relevant law article community in the construction of the graph, and capture information at a variety of granularities (local contextual and semantic information). LADAN directly inputs the fact description text sequences into the text encoder. 2. differences of backbone networks: a. LA-MGFM uses graph neural network as the backbone network, and the fact description text is processed into text graph, and then the information is interactively processed by the graph neural network. LADAN uses Bi-GRU as the backbone network, and only uses the graph neural network to get the vector representation of the law article community. b. LA-MGFM designs novel intra-graph interaction and inter-graph interaction methods. In the intra-graph interaction method, we incorporate sememe information to enhance the graph neural network for modeling word and semantics. In the inter-graph interaction, we design an adequate multi-graph fusion mechanism that can be well adapted to the five text graphs constructed in this paper. In contrast, LADAN uses only bidirectional GRU networks in text modeling. Furthermore, existing work lacks attention in low resource scenarios, which we think that cannot be ignored. Thus, we perform our approach in few-shot condition and consider the construction of different text graphs as a means of graph data augmentation. The performance improvement is still obvious in the few-shot setting, revealing the potential of combining different text graphs and knowledge of law article in low-resource scenarios.

6.2. Contributions to related research

The aim of this paper is to offer a new perspective on LJP through the use of graph neural networks. LJP is a well-known classification task that is ideally suited to leveraging GNNs for both node and graph classification. Our work is the first to construct multiple textual graphs and utilize intra- and inter-graph information for LJP. As such, our approach can serve as a robust new baseline for future research. At present, the research on utilizing GNNs in NLP is still in its early stages. However, once we have confirmed the efficacy of GNNs in LJP, we can delve into more advanced GNNs-based techniques for this purpose. Our approach, which involves integrating knowledge from both sememe and law articles into the model, is versatile and can be applied to other tasks as well. We are confident that our method's exceptional performance in the typical classification task of LJP makes it worth exploring in other classification tasks. Moreover, our study showcases the effectiveness of utilizing multi-graph fusion techniques to extract detailed information from multiple text graphs. This provides a strong foundation for future research in this area.

6.3. Limitations and future work

There is room for improvement in our work. First, it is important to note that different datasets possess unique characteristics, which necessitates the development of a viable method for achieving generalizability. Additionally, the proposed multiple graph fusion approach is straightforward and does not require extensive mining knowledge to guide the key graph selection process in any inter-graph fusion mechanism. Furthermore, the method is particularly suitable for interpretation in the legal domain, and future work should focus on devising an interpretable analysis strategy for the approach. Several research have now demonstrated that the LJP task's performance can be enhanced by incorporating information extraction from the domain. Additionally, the relevant law article utilized in this study can be viewed as structured external knowledge, supplementing the fact description text. Future work could involve the inclusion of other forms of knowledge, such as structured text or images. Finally, our approach uses Lawformer to obtain word vectors, a model that is good at processing long texts, but may not optimal for short texts. We plan to add improvements to Lawformer in future work to enhance its ability to handle short texts.

7. Conclusion

In this paper, we present a new approach (LA-MGFM) to legal judgment prediction that utilizes a sememe-enhanced GGNN (SE-GGNN) and a multi-graph fusion mechanism (M-Fusion) to address the problem of charge confusion by incorporating information from law articles. Our method is unique in that it constructs five different text graphs and employs a multi-graph fusion mechanism, which has not been proposed before for LJP. Our approach, LA-MGFM, outperforms previous methods in both few-shot and full data conditions. On the one hand, we merge fact descriptions with pertinent law articles to create various text graphs that encompass both local contextual and semantic information. On the other hand, we incorporate a fine-grained intra-graph information interaction and inter-graph fusion mechanism to further enhance our model. In order to evaluate our approach, we conduct experiments on two benchmark datasets. To provide a fair comparison, we select machine learning and deep learning methods, some of which also incorporated external knowledge of law articles. Our approach outperform the comparison models in terms of accuracy, precision, recall, and macro F1 values, demonstrating its significant advantage. our experiments have demonstrated the significant advantages of constructing and fusing multiple graphs for the LJP task, ultimately leading to new state-of-the-art results. Finally, we have conducted a thorough analysis to gain a comprehensive understanding of our approach.

In our future research, we aim to apply the current method to other LegalAI tasks. Furthermore, we plan to investigate a more efficient method for constructing graphs in natural languages such as text. Additionally, we will work towards developing an improved interpretable multi-graph fusion method that can effectively and comprehensively merge information from nodes and edges in various types of graphs.

CRediT authorship contribution statement

Qihui Zhao: Conceptualization of this study, Methodology, Software, Formal analysis, Writing – original draft, Data curation. **Tianhan Gao:** Supervision, Writing – review & editing. **Nan Guo:** Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by National Natural Science Foundation of China (grant no. 52130403) and the Fundamental Research Funds for the Central Universities (grant no. N2017003).

References

Chen, et al. (2019). Charge-based prison term prediction with deep gating network. In K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 6361–6366). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D19-1667.

Chen, et al. (2020). Iterative deep graph learning for graph neural networks: Better and robust node embeddings. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), *Advances in neural information processing systems 33: annual conference on neural information processing systems 2020*. URL https://proceedings.neurips.cc/paper/2020/hash/e05c7ba4e087beea9410929698dc41a6-Abstract.html.

Dai, et al. (2022). Graph fusion network for text classification. *Knowledge-Based Systems*, *236*, Article 107659.

Defferrard, et al. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, R. Garnett (Eds.), *Advances in neural information processing systems 29: annual conference on neural information processing systems 2016, December 5-10, 2016, Barcelona, Spain* (pp. 3837–3845). URL https://proceedings.neurips.cc/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html.

Dong, & Dong (2003). HowNet-a hybrid language and knowledge resource. In *NLP-KE*.

Gilmer, et al. (2017). Neural message passing for quantum chemistry. In D. Precup, Y. W. Teh (Eds.), *Proceedings of Machine Learning Research*: *70*, *Proceedings of the 34th International conference on machine learning* (pp. 1263–1272). PMLR, URL http://proceedings.mlr.press/v70/gilmer17a.html.

Hamilton, et al. (2017). Inductive representation learning on large graphs. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), *Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, December 4-9, 2017, Long Beach, CA, USA* (pp. 1024–1034). URL https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html.

Hu, et al. (2018a). Few-shot charge prediction with discriminative legal attributes. In E. M. Bender, L. Derczynski, P. Isabelle (Eds.), *Proceedings of the 27th International conference on computational linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018* (pp. 487–498). Association for Computational Linguistics, URL https://aclanthology.org/C18-1041/.

Hu, et al. (2018b). Few-shot charge prediction with discriminative legal attributes. In E. M. Bender, L. Derczynski, P. Isabelle (Eds.), *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018* (pp. 487–498). Association for Computational Linguistics, URL https://aclanthology.org/C18-1041/.

Hu, et al. (2019). Heterogeneous graph attention networks for semi-supervised short text classification. In K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), *Proceedings of the 2019 Conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 4820–4829). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D19-1488.

Huang, et al. (2019). Text level graph neural network for text classification. In K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), *Proceedings of the 2019 Conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 3442–3448). Association for Computational Linguistics.

Jiang, et al. (2018). Interpretable rationale augmented charge prediction system. In D. Zhao (Ed.), *COLING 2018, the 27th International conference on computational linguistics: system demonstrations, Santa Fe, New Mexico, August 20-26, 2018* (pp. 146–151). Association for Computational Linguistics, URL https://aclanthology.org/C18-2032/.

Kang, et al. (2019). Creating auxiliary representations from charge definitions for criminal charge prediction. CoRR abs/1911.05202 arXiv:1911.05202.

Katz, et al. (2014). A general approach for predicting the behavior of the supreme court of the united states. *PLoS One*, *12*(4), Article e0174698.

Kim (2014). Convolutional neural networks for sentence classification. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 Conference on empirical methods in natural language processing* (pp. 1746–1751). ACL, http://dx.doi.org/10.3115/v1/d14-1181.

Kingma, & Ba (2015). Adam: A method for stochastic optimization. In Y. Bengio, & Y. LeCun (Eds.), *3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL http://arxiv.org/abs/1412.6980.

Kipf, & Welling (2017). Semi-supervised classification with graph convolutional networks. In *5th International conference on learning representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, URL https://openreview.net/forum?id=SJU4ayYgl.

Kort, F. (1957). Predicting supreme court decisions mathematically: A quantitative analysis of the right to counsel cases. *American Political Science Review*, *51*(1), 1–12.

Lai, et al. (2015). Recurrent convolutional neural networks for text classification. In B. Bonet, S. Koenig (Eds.), *Proceedings of the twenty-ninth AAAI conference on artificial intelligence, January 25-30, 2015, Austin, Texas, USA* (pp. 2267–2273). AAAI Press, URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9745.

Li, et al. (2019). MANN: A multichannel attentive neural network for legal judgment prediction. *IEEE Access*, *7*, 151144–151155. http://dx.doi.org/10.1109/ACCESS.2019.2945771.

Liu, & Hsieh (2006). Exploring phrase-based classification of judicial documents for criminal charges in Chinese. In F. Esposito, Z. W. Ras, D. Malerba, & G. Semeraro (Eds.), *Lecture Notes in Computer Science*: *vol. 4203*, *Foundations of intelligent systems, 16th international symposium, ISMIS 2006, Bari, Italy, September 27-29, 2006, Proceedings* (pp. 681–690). Springer, http://dx.doi.org/10.1007/11875604_75.

Liu, et al. (2015). Predicting associated statutes for legal problems. *Inf. Process. Manag.*, *51*(1), 194–211. http://dx.doi.org/10.1016/j.ipm.2014.07.003.

Liu, et al. (2019). Legal cause prediction with inner descriptions and outer hierarchies. In M. Sun, X. Huang, H. Ji, Z. Liu, Y. Liu (Eds.), *Lecture Notes in Computer Science*: *11856*, *Chinese computational linguistics - 18th China national conference* (pp. 573–586). Springer, http://dx.doi.org/10.1007/978-3-030-32381-3_46.

Liu, et al. (2020). Tensor graph convolutional networks for text classification. In *The Thirty-Fourth AAAI Conference on artificial intelligence, AAAI 2020, the Thirty-Second innovative applications of artificial intelligence conference, IAAI 2020, the Tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020* (pp. 8409–8416). AAAI Press, URL https://ojs.aaai.org/index.php/AAAI/article/view/6359.

Long, et al. (2019). Automatic judgment prediction via legal reading comprehension. In M. Sun, X. Huang, H. Ji, Z. Liu, Y. Liu (Eds.), *Lecture Notes in Computer Science*: *vol. 11856*, *Chinese computational linguistics - 18th China national conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings* (pp. 558–572). Springer, http://dx.doi.org/10.1007/978-3-030-32381-3_45.

Luo, et al. (2017). Learning to predict charges for criminal cases with legal basis. In M. Palmer, R. Hwa, S. Riedel (Eds.), *Proceedings of the 2017 Conference on empirical methods in natural language processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017* (pp. 2727–2736). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/d17-1289.

Mikolov, T., et al. (2013). Efficient estimation of word representations in vector space. In Y. Bengio, Y. LeCun (Eds.), *1st International conference on learning representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. URL http://arxiv.org/abs/1301.3781.

Minaee, et al. (2021). Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys*, *54*(3), 62:1–62:40. http://dx.doi.org/10.1145/3439726.

Nagel, S. (1964). Applying correlation analysis to case prediction. *Texas Law Review*, *42*(7), 1006–1017.

Niu, et al. (2017). Improved word representation learning with sememes. In R. Barzilay, M. Kan (Eds.), *Proceedings of the 55th annual meeting of the association for computational linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers* (pp. 2049–2058). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P17-1187.

Pan, et al. (2019). Charge prediction for multi-defendant cases with multi-scale attention. In Y. Sun, T. Lu, Z. Yu, H. Fan, L. Gao (Eds.), *Communications in Computer and Information Science*: *vol. 1042*, *Computer supported cooperative work and social computing - 14th CCF Conference, ChineseCSCW 2019, Kunming, China, August 16-18, 2019, Revised Selected Papers* (pp. 766–777). Springer, http://dx.doi.org/10.1007/978-981-15-1377-0_59.

Pennington, et al. (2014). Glove: Global vectors for word representation. In A. Moschitti, B. Pang, W. Daelemans (Eds.), *Proceedings of the 2014 Conference on empirical methods in natural language processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, a Meeting of SIGDAT, a Special Interest Group of the ACL* (pp. 1532–1543). ACL, http://dx.doi.org/10.3115/v1/d14-1162.

Qi, et al. (2019). OpenHowNet: An open sememe-based lexical knowledge base. CoRR abs/1901.09957 arXiv:1901.09957.

Qin, et al. (2020). Improving sequence modeling ability of recurrent neural networks via sememes. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 2364–2373.

Shervin, et al. (2021). Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys*, *54*(3), 62:1–62:40.

Suykens, & Vandewalle (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293–300. http://dx.doi.org/10.1023/A:1018628609742.

Wang, et al. (2019). Using case facts to predict accusation based on deep learning. In *19th IEEE International conference on software quality, reliability and security companion, QRS Companion 2019, Sofia, Bulgaria, July 22-26, 2019* (pp. 133–137). IEEE, http://dx.doi.org/10.1109/QRS-C.2019.00038.

Wei, & Lin (2019). An external knowledge enhanced multi-label charge prediction approach with label number learning. CoRR abs/1907.02205 arXiv:1907.02205.

Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., et al. (2018). CAIL2018: a large-scale legal dataset for judgment prediction. CoRR abs/1807.02478 arXiv:1807.02478.

Xiao, et al. (2021). Lawformer: A pre-trained language model for Chinese legal long documents. *AI Open*, [ISSN: 2666-6510] *2*, 79–84. http://dx.doi.org/10.1016/j.aiopen.2021.06.003, URL https://www.sciencedirect.com/science/article/pii/S2666651021000176.

Xu, et al. (2020a). Distinguish confusing law articles for legal judgment prediction. In D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), *Proceedings of the 58th Annual meeting of the association for computational linguistics, ACL 2020, Online, July 5-10, 2020* (pp. 3086–3095). Association for Computational Linguistics.

Xu, et al. (2020b). Distinguish confusing law articles for legal judgment prediction. In D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), *Proceedings of the 58th Annual meeting of the association for computational linguistics, ACL 2020, Online, July 5-10, 2020* (pp. 3086–3095). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.280.

Yang, et al. (2016). Hierarchical attention networks for document classification. In K. Knight, A. Nenkova, O. Rambow (Eds.), *NAACL HLT 2016, the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016* (pp. 1480–1489). The Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/n16-1174.

Yang, et al. (2019). Legal judgment prediction via multi-perspective bi-feedback network. In S. Kraus (Ed.), *Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI 2019, Macao, China, August 10-16, 2019* (pp. 4085–4091). ijcai.org, http://dx.doi.org/10.24963/ijcai.2019/567.

Yao, et al. (2019). Graph convolutional networks for text classification. In *The Thirty-Third AAAI Conference on artificial intelligence, AAAI 2019, the thirty-first innovative applications of artificial intelligence conference, IAAI 2019, the Ninth AAAI Symposium on educational advances in artificial intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019* (pp. 7370–7377). AAAI Press.

Zhang, et al. (2020a). Every document owns its structure: Inductive text classification via graph neural networks. In D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), *Proceedings of the 58th Annual meeting of the association for computational linguistics, ACL 2020, Online, July 5-10, 2020* (pp. 334–339). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.31.

Zhang, et al. (2020b). A practical Chinese dependency parser based on a large-scale dataset. CoRR abs/2009.00901 arXiv:2009.00901.

Zhang, et al. (2021). Multi-modal graph fusion for named entity recognition with targeted visual guidance. In *Thirty-Fifth AAAI Conference on artificial intelligence, AAAI 2021, Thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the Eleventh symposium on educational advances in artificial intelligence, EAAI 2021, Virtual Event, February 2-9, 2021* (pp. 14347–14355). AAAI Press, URL https://ojs.aaai.org/index.php/AAAI/article/view/17687.

Zhao, et al. (2022). A novel chinese relation extraction method using polysemy rethinking mechanism. *Applied Intelligence*.

Zhong, et al. (2018). Legal judgment prediction via topological learning. In E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii (Eds.), *Proceedings of the 2018 Conference on empirical methods in natural language processing, Brussels, Belgium, October 31 - November 4, 2018* (pp. 3540–3549). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/d18-1390.