International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 1632-1649

https://doi.org/10.70517/ijhsa464144

Intelligent Classification Algorithm Model for Legal Documents in the Digital Rule of Law Framework

Chun Ren[1,*], Panpan Li[2] and Yuhui Lei[1]
[1] Zhengzhou Urban Construction Vocational College, Zhengzhou, Henan, 451263, China
[2] Henan Technical College of Construction, Zhengzhou, Henan, 450064, China
Corresponding authors: (e-mail: 13783588878@163.com).

Abstract In recent years, with the booming development of deep learning, it makes the judicial big data analysis technology increasingly attracts people's attention, and gives rise to many new applications oriented to the judicial field, and the intelligent court project is also included in the judicial construction process. The research realizes the automatic classification and retrieval model of legal documents through the graph neural network-based algorithm, and the article proposes the DASA-GNN text classification model based on the Texting model, and adopts the EDA data enhancement method combined with the self-attention mechanism in the data sampling stage. Meanwhile, a graph convolutional network is introduced on the basis of the pre-trained model to learn the contextual information and global structured information of the text, and a text retrieval model based on the topological feature representation of the convolutional graph is proposed. Then the application effect of the classification model as well as the retrieval model is analyzed through experiments, and the results show that the Acc value of the DASA-GNN text classification model reaches 95.12% on the PKULawData dataset, which is improved compared with that of the baseline models such as BERT, DADGNN, and RCNN, etc.; finally, a comparison experiment is carried out on the legal retrieval dataset LeCaRD, and the experimental The results show that the text retrieval model based on the topological feature representation of convolutional graph has a better retrieval effect compared to other retrieval methods.

Index Terms attention mechanism, legal text categorization, graph neural network, graph convolutional network

I. Introduction

Referee documents are legal documents that record the process and results of the people's court proceedings, the official provisions of the unified writing norms [1]. Referee documents to the parties and society to show the reasonableness, legality, impartiality, finality of the conclusions of the referee, is the judicial process of refining and summarizing, is the crystallization of the results of the trial, is an important carrier and final embodiment of judicial justice [2]-[4]. The supreme people's court unified production of the style of decision-making documents, for the country's four courts and the majority of judges to provide a standardized text of the one to follow, is not only the inevitable requirements of strict and fair justice, but also judicial activities, judicial behavior standardization, publicity of the best embodiment of the [5]-[7]. Due to the high degree of similarity of the case judgment results are different, will result in the consequences of violating the fairness and impartiality of justice, reducing the credibility of justice [8], [9]. Therefore, in order to better implement the notice of the Supreme Court and better help judges to quickly and accurately get the historical cases with high similarity to the current case, the study of intelligent case-like retrieval has important social significance and application value [10]-[13].

Once the case handlers of various organizations faced with a huge amount of legal documents can not help, one by one, the time cost of the trial is very high, consuming energy at the same time, the ensuing accumulation of fatigue may have a certain impact on the processing of the next case [14]-[16]. For the case of automatic classification and retrieval of the advantages of this time appears to be particularly prominent, not only can speed up the case handler to the case of the relevant legal documents reading speed, but also can accelerate the understanding of the case [17], [18]. In the face of the increasing amount of case text in the judicial field, people have gradually begun to join the application of natural language processing technology in the processing [19]. Element extraction is a fundamental task of natural language processing to extract structured knowledge from unstructured text that can be understood by machines or programs. Literature [20] describes an information extraction system applicable to legal documents, i.e., the AIS system is able to automatically acquire structured data from legal documents, eliminating consistency errors in the extracted data after the introduction of a knowledge ontology, and providing high-quality information extraction results for subsequent legal analysis. Literature [21] establishes LegalVis, a web-based visual analysis system for legal cases, which strengthens the reliability and professionalism of the cited legal cases by visually exploring legal precedents with constraints and constructing relevant classification models and interpretation models. Literature [22] explores the application of automated tools in the problem of legal case retrieval by developing a semi-automated literature retrieval system based on text mining, natural language processing and data visualization methods, which enables the retrieval and analysis of similar cases and the formation of preferred reports. Literature [23] shows that a reliable annotated corpus is an important prerequisite for the exploration of legal documents, and proposes a named entity recognition method based on conditional random fields and bi-directional long and short-term memory networks to construct a corpus applicable to legal documents, which can support the retrieval task in similar contexts. Literature [24] utilizes entity recognition (NER) and lexical annotation (POS) methods for element extraction of legal documents and embeds a large-scale language model to improve the effectiveness and accuracy of full-text retrieval of legal documents. Literature [25] evaluates the effectiveness of GPT-4 as a large-scale language model in key information extraction of legal judgments, showing that its automatic information extraction capability can accurately describe the complexity of the case and the rationale behind the judicial decision, providing valuable references for judicial practice. However, the work in the above literature on the recommendation of class cases for adjudication documents is very limited, and the studies on the retrieval of class cases are based on the content vectorization of a whole legal text to retrieve the class cases, which does not combine with the characteristics of legal expertise, and the effect is not good. Therefore, graph algorithms can be introduced to focus on the differences and complexity between knowledge nodes and adaptively learn the relationship weights between the nodes, so as to better model the complex knowledge relationships in graph data.

In this paper, the automatic classification and retrieval model of legal documents are realized based on graph neural network algorithm respectively. Firstly, the DASA-GNN model is proposed, which retains the graphical coding characteristics in graph neural networks while using data enhancement techniques, enabling the model to maintain relatively stable performance even with a small amount of training data. Further, the model strengthens the interconnections at the word level in the original text by introducing a self-attention mechanism, and at the same time enhances the extraction of textual information at the sentence level, which effectively improves the overfitting problem of graph neural networks in text categorization tasks. Then a text retrieval model based on convolutional graph topology feature representation is proposed to generate new text features using graph convolutional network after converting the context vectors of query and document into graph topology. This type of text feature representation contains not only contextual information, but also global structured information. Finally, the accuracy rate of automatic classification and retrieval of legal documents is tested experimentally.

II. Automatic classification and retrieval model for legal documents based on graph algorithms

II. A. Neural network model based on graph algorithm

II. A. 1) Types of diagram structures

A graph structure is an abstract data structure used to represent relationships between objects. Typically a graph structure is represented by $G=(V,E)$, V represents the set of nodes of the graph, and E represents the set of edges of the graph. Graph structures can be used to model a variety of real-world problems such as social networks, network topologies, routing algorithms, and so on. According to the nature of the nodes and edges in the graph, graph structures can be divided into various types, and the common types of graph structures are undirected graphs, directed graphs, weighted graphs, heterogeneous graphs, isomorphic graphs, and so on.

II. A. 2) Graph Convolutional Neural Networks

Graph Convolutional Neural Networks (GCNs) are mainly categorized into two main types: spectral domain and null domain. The core concept of spectral domain GCNs is derived from the spectral theory of graphs, where the convolution kernel is constructed from the feature vectors of the Laplace matrix of the graph and in this way the convolution operations on the graph are mapped into the frequency domain, thus performing convolution in the spectral domain of the graph [26]. The basic idea of null domain GCNs is to use the neighbor information of the nodes for feature extraction and convolution operations.

(1) Spectral domain graph convolutional neural network

Spectral domain graph convolution operation enables feature extraction and representation learning on graphs by converting the node feature matrix to frequency domain for convolution operation through feature decomposition of graph Laplacian matrix. Given a graph $G=(V,E,A)$, V is the set of nodes, assuming there are n nodes, $A\in R^{n\times n}$ is the adjacency matrix, each node also has d-dimensional features, and $x\in R^{n\times d}$ is the feature vector matrix of the nodes. Usually the Laplacian matrix is defined as $L = D - W$, and the normalized Laplacian matrix is $L = I_n - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. By eigenvalue decomposition, the Laplacian matrix L can be represented as $L = U \wedge U^T$, where U is the eigenmatrix of eigenvectors and $\wedge$ is the diagonal matrix of eigenvalues. In the normalized Laplace matrix, the eigenvectors are orthogonal to each other, i.e., $UU^T = 1$. In addition, the Fourier transform of the graph plays a key role in graph convolutional networks. Assuming the existence of a graph signal $x \in R^n$, the Fourier transform of the graph signal is $F(x) = U^T x$ and the inverse transform is $F^{-1}(x) = Ux$. The graph Fourier transform projects the input graph signal into an orthogonal space, where the basis of the orthogonal space consists of the eigenvectors of the normalized Laplacian operator.

(2) Convolutional Neural Networks for Nullspace Maps

Null domain graph convolutional neural networks are suitable for processing graph data with spatial attributes, such as maps, social networks, and molecular structures. Unlike graph convolutional networks based on spectral theory, null domain graph convolutional networks operate directly on the nodes and edges of the graph, rather than processing through the spectral space of the graph.

II. A. 3) Graph Attention Neural Networks

Unlike traditional Graph Convolutional Networks (GCNs), GAT employs an attention mechanism to characterize the interactions between nodes and adaptively integrates the information from neighboring nodes by automatically learning the weights between these nodes.The core idea of the GAT model is to determine the importance of a node with respect to its neighbors through an attention mechanism, which in turn weights and aggregates the features of the neighboring nodes to generate a more information-rich node representation.

Specifically, for node v, its attention score e_{uv} is calculated by weighting with its neighbor node u, and then the features of the neighbor nodes are multiplied by the corresponding attention weights and normalized for aggregation. This can make the model pay more attention to the neighbor nodes that have important contributions to the node representation, so as to carry out information transfer and feature update more effectively. The specific computational expression is as follows:

$$e_{uv} = Leaky\,\mathrm{Re}\,LU\left(\vec{a}^T \left[Wh_v \| Wh_u\right]\right) \tag{1}$$

$$\alpha_{uv} = soft\max_u \left(e_{uv}\right) \tag{2}$$

$$h_v' = \sigma\left(\sum_{u \in N(v)} \alpha_{uv} W h_u\right) \tag{3}$$

where, h_v and h_u denote the feature representation of node v and neighbor node u respectively, $\vec{a}$ is the attention weight vector, W is the weight matrix, σ is the activation function, and $soft\max_u(\cdot)$ is the normalization operation on neighbor nodes. h_v is the feature-weighted summation of the attention coefficients. In order to enhance the model's representational ability and stability, GAT introduces a multi-head attention mechanism, as shown in Equation (4):

$$h_v' = \|_{k=1}^{K} \sigma\left(\sum_{u \in N(v)} \alpha_{uv}^k W^k h_u\right) \tag{4}$$

where $\|$ denotes the vector splicing operation.The GAT model is able to learn the representation of nodes efficiently and achieves excellent performance on graph data. Due to its introduction of the attention mechanism, GAT has enhanced representational capabilities and flexibility in learning graph structural information and relationships between nodes, providing a powerful tool and methodology for solving various graph data challenges.

II. A. 4) Graph Gating Neural Networks

GGNN employs a gating mechanism similar to that in RNN to control the information transfer and update process. The representation of nodes is dynamically adjusted by introducing update gates and reset gates to better capture complex dependencies in the graph data. This gating mechanism enables the GGNN to flexibly adjust the information transfer and aggregation process according to the state of the current node and the information of the neighboring nodes in each iteration step, thus improving the model's representational ability and adaptability.

The main idea of the GGNN model is to utilize the idea of gated recurrent unit (GRU) to deal with graph structure data. In the graph, each node is represented as a vector and the state of the node is dynamically updated by a recurrent neural network. In each iteration step, the GGNN first uses an update gate to determine whether to update the node's representation based on the current state and the information of the neighboring nodes, and then uses a reset gate to control how to merge the node's current representation and the information of the neighboring nodes. This gating mechanism effectively captures long-distance dependencies between nodes and allows the model to flexibly adjust the information transfer and aggregation process at each iteration step. This is shown in Eqs. (5), (6) and (7):

$$z_i^{(t)} = \sigma\left(W_z \cdot \left[h_i^{(i-1)}, m_i^{(t)}\right] + b_z\right) \tag{5}$$

$$r_i^{(t)} = \sigma\left(W_r \cdot \left[h_i^{(i-1)}, m_i^{(t)}\right] + b_r\right) \tag{6}$$

$$\tilde{h}_i^{(t)} = \tanh\left(W_h \cdot \left[r_i^{(i)} \square\ h_i^{(i-1)}, m_i^{(i)}\right] + b_h\right) \tag{7}$$

where $z_i^{(i)}$ is the update gate of node i at moment t, $r_i^{(i)}$ is the reset gate of node i at moment t, $\tilde{h}_i^{(i)}$ is the new state of node i at moment t, the new node representation obtained by the gating mechanism. σ is the activation function, $m_i^{(t)}$ is the information of the neighbor node i at the current moment t, which is usually an aggregation of the representation vectors of the neighbor nodes.

II. B.Representation of the text of legal instruments

Text representation is the conversion of more human understandable text into a machine understandable representation of numerical vectors. In the earlier period, bag-of-words model is a commonly used method for text feature representation. The bag-of-words model takes all the words and word frequencies in the text as text representation vectors, i.e., the text is converted into feature vectors, and each dimension in the feature vectors represents a word or phrase in the vocabulary, and the importance of the word or phrase is expressed by calculating the frequency of its occurrence in the text.

The TFIDF model improves the bag-of-words model by being able to filter common words and highlight the weights of important words. Lexical item frequency indicates the importance of a word (or phrase) in the text by the frequency of its occurrence in the text, the higher the frequency means the more important the word is in the text. Inverse Document Frequency measures the general importance of a word in the entire text collection, the larger the IDF value, the less common the word is in the entire text collection and the more representative of the document in which the word is found.TFIDF is a combination of TF and IDF to calculate the importance of each word in the text corpus. For a given word and document, by calculating the TFIDF value of each word in the text, the importance of the word is evaluated according to the frequency and prevalence of the word, and the feature vector of each document is obtained, which can distinguish the differences between different texts, with the following formula:

$$TFIDF_{x,d} = TF_{x,d} \times IDF_x \tag{8}$$

$$TF_{x,d} = \frac{\begin{gathered}Number\ of\ occurrences\ of\\ word\ x\ in\ document\ d\end{gathered}}{Total\ number\ of\ words\ in\ d} \tag{9}$$

$$IDF_x = \log\left(\frac{Total\ number\ of\ documents}{\begin{gathered}Number\ of\ documents\\ containing\ the\ word\ x\end{gathered}}\right) \tag{10}$$

Doc2vec is an unsupervised vectorized representation model of documents 2%', which is based on the Word2vec model.A common training method for Doc2vec is PV-DM (Distributed Memory model of Paragraph Vectors), as shown in Figure 1. Multiple context words in a paragraph are used as inputs to the neural network, and one target word is used as the neural network output, which can be called the “many-to-one” training method. The model defines a sliding window to fetch words in a paragraph, so that the number of words input to the model is fixed for each training session. For a language like English, a typical window size is 10, i.e., 10 words are used as context words for a target word.

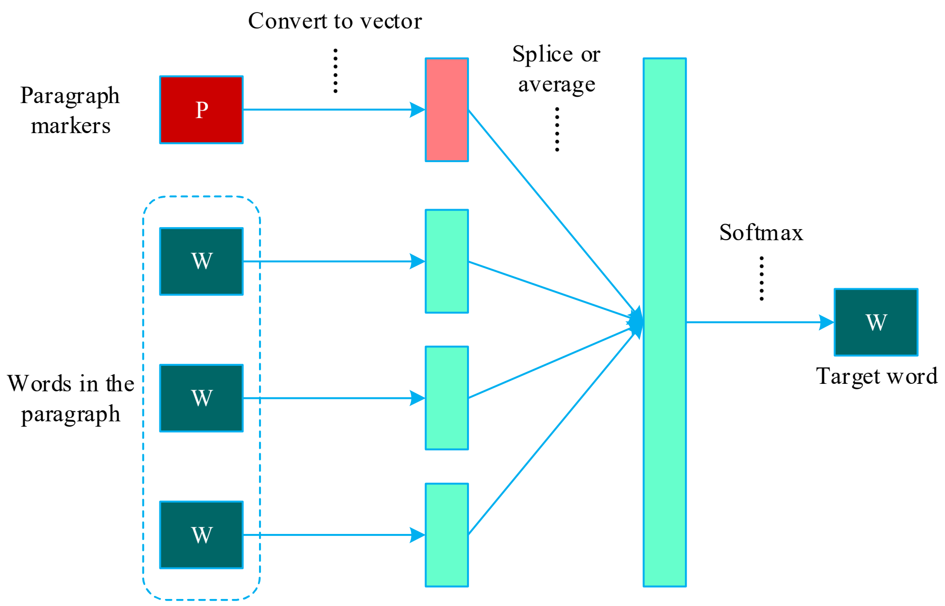


Figure 1: Doc2vec's pv-dm model

Doc2vec has one more paragraph markup vector than Word2vec, which can also be called document markup vector. Each paragraph shall be assigned an identification number when input to the neural network and packed with all the words of the paragraph, so the actual size of the window is the number of input context words plus 1. Texts in the same paragraph all have the same paragraph marking vector during training, and are thus called distributed memory models of paragraph vectors.Word2vec ignores the word order of the text, while Doc2vec, because of the shared paragraph marker vectors, all the information of sliding to fetch words is then incorporated into the paragraph marker vectors. Therefore, Doc2vec has no restriction on document length and is sensitive to the word order of the input.

The Doc2vec model indicates that the goal of learning is not to obtain the result of the output layer, but the result of the compression of the information of the input layer by the hidden layer in the middle of the model, which is known as embedding, a process of mapping high-dimensional sparse vectors to low-dimensional dense vectors, and the number of neurons in the hidden layer is the number of embedding features. After training, the weights of the hidden layer neurons contain the model's knowledge of the input information, and the process of training is the process of the neurons learning the weight matrix, similar to adjusting the coefficients in front of each term when fitting a polynomial. The output layer neurons use Softmax regression classifiers, and the number of Softmax classifiers in the output layer is determined by the number of categories to be categorized. Each Softmax classifier gives the normalized prediction probability of all words in the glossary, and the word with the highest probability is taken as the target word for prediction.

II. C.Text categorization model based on DASA-GNN

In the deep learning text classification task, the new model graph neural network GNN has achieved good results. Compared with the traditional model, GNN has more advantages in text data processing and classification accuracy. However, the graph-text conversion feature of GNN causes it to be prone to overfitting when the training dataset is small. To address this problem, this chapter proposes the DASA-GNN model based on Texting model and data enhancement methods. Figure 2 shows the block diagram of the structure of the DASA-GNN model.

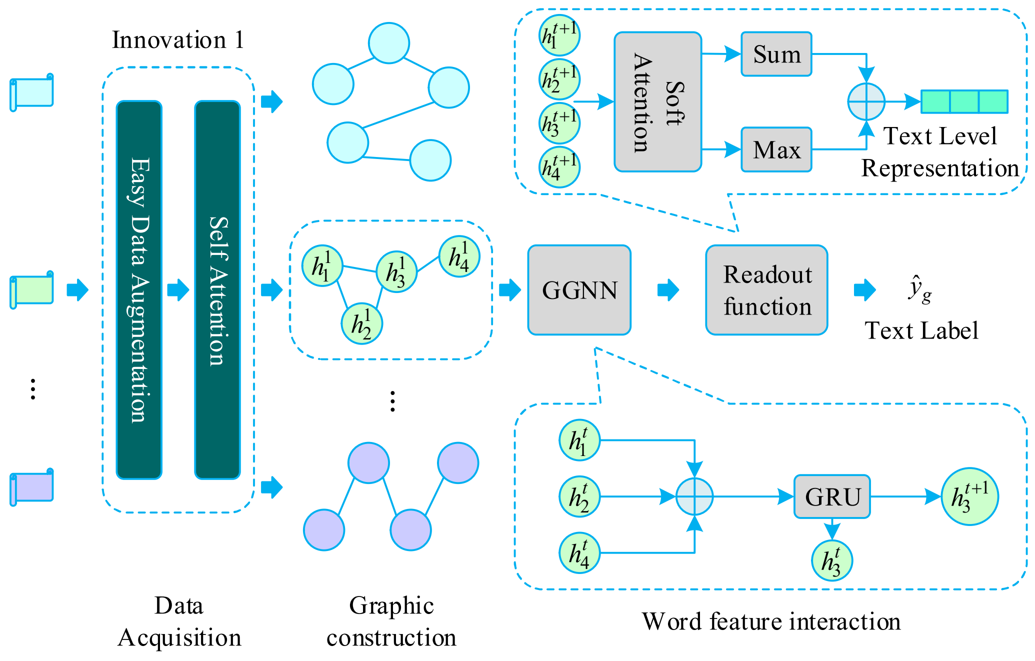


Figure 2: DASA-GNN model

DASA-GNN consists of four main parts: the first part is the data-enhanced data sampling part generated by the combination of the self-attention mechanism and EDA. The second part is graph construction using sliding windows. The third part is word feature interaction based on gated graph neural network. Finally, the extracted features are fed into two multilayer perceptron machines to complete the predictive classification of the text.

II. C. 1) Data sampling

(1) Synonym Replacement: randomly selecting the words in a sentence and replacing them synonymously with close analogs of those words to form a new sentence. For example, “I'm going to graduate soon” is changed to “I'm going to finish my studies soon” by extracting the word “graduate” and replacing it with a similar word using synonyms. This type of noise addition makes the semantics of the sentences not change in polarity, and there is a high probability that the final classification results are consistent.

(2) Random insertion: a word is randomly selected in a sentence, after which a synonym of the word is randomly inserted into any position in the sentence.

(3) Random exchange: two words selected at random in a sentence are interchanged in position.

(4) Random deletion: any word in a sentence is probabilistically deleted randomly with probability p.

After using EDA for data enhancement of text, we can get effective data that is several times more than the original data. When performing the EDA operation, we define the sentence length as l, a percentage parameter as α, and the amount of word change as $n = l\alpha$.

Next, the DASA-GNN model introduces a self-attention mechanism after the EDA data enhancement. The purpose of this is to allow the extracted features to have more details.

The key parameters of the self-attention mechanism can be expressed as equation (11).

$$Att(Q,K,V) = \omega\left(QK^T\right)V \tag{11}$$

where $Att(Q,K,V)$ is the value of the obtained attention, Q is the query vector matrix, K is the key vector matrix, and V is the value vector matrix. Each row of these three matrices corresponds to a vector representation, respectively. This vector representation is generally obtained by multiplying the input sequence by the three matrices $W^{(q)}$, $W^{(k)}$, and $W^{(v)}$, respectively.

II. C. 2) Graphic construction

The graph construction section in Figure 2 shows that first the selected words in a sentence are represented as vertices, and then the graph is constructed using co-occurrences between words in the form of edges, which can be represented as $G = (V, E)$, where V represents the vertices of the graph and E represents the edges of the graph. Co-occurrence refers to the correlation of words in a sliding window, where the sliding window size is generally set to 3 by default, and the edges in it are all undirected edges.Nikolentzos et al. define the size of the sliding window to be 2. They consider graphs as densely connected graphs, and the mechanism of graph message delivery in their model is mainly to connect a particular basic node to every other node, so that in the graph we can only get fuzzy structural messages. While in the gated graph neural network in order to avoid the blurring of word feature information caused by the dense connection of the graph, it will first initialize the word features of the text data to perform the embedding representation of the vertices, and then any one of the documents is represented as a separate subgraph, so that in the part of the word interaction phase in the model, the word feature information can be clearly propagated into the context.

II. C. 3) Graph-based word feature interaction

At the end of the graph construction, the embedded representation of the word nodes can be learned to be represented by a gated graph neural network.GGNN is characterized by the fact that any of its nodes can use the edges to receive information from its neighboring nodes, and then update the information by connecting the received information with the information it represents itself. Gated graph neural networks mainly use the adjacency matrix for information updating.

The recursive formulas for the propagation model are shown in Eq. (12) to Eq. (17).

$$h_v^{(1)} = \left[x_v^T, 0\right]^T \tag{12}$$

$h_v^{(1)}$ in Equation (12) is the initial state of node v and is a D-dimensional vector. It indicates that when the node input feature x_v dimension is less than D, it is complemented with 0 at the end.

$$a_v^{(t)} = A_v^T \left[h_1^{(t-1)T} \cdots h_v^{(t-1)T}\right]^T + b \tag{13}$$

In Equation (13) $A_{v:}^T$ denotes the first two columns of the matrix A corresponding to the selected node v, followed by $\left[h_1^{(t-1)T} \cdots h_v^{(t-1)T}\right]^T$ which denotes the matrix formed by augmenting all the node features at the moment $t-1$. $a_v^{(t)}$ denotes the result obtained after updating by edges between the current node and its neighboring nodes.

$$z_v^t = \sigma\left(W^z a_v^{(t)} + U^z h_v^{(t-1)}\right) \tag{14}$$

$$r_v^t = \sigma\left(W^z a_v^{(t)} + U^\tau h_v^{(t-1)}\right) \tag{15}$$

$$\tilde{h}_v^{(t)} = \tanh\left(W a_v^{(t)} + U\left(r_v^t \odot h_v^{(t-1)}\right)\right) \tag{16}$$

$$h_v^{(t)} = \left(1 - z_v^t\right) \odot h_v^{(t-1)} + z_v^t \odot \tilde{h}_v^{(t)} \tag{17}$$

Eqs. (14) through (17) are similar to the GRU computational process. z_v^t controls for forgotten information and r_v^t controls for newly generated information. Eq. (16) in $r_v^t \odot$ determines from which past information new information is generated. $\tilde{h}_v^{(t)}$ is the newly generated information. $h_v^{(t)}$ is the final updated node state. In Equation (17) $\left(1 - z_v^t\right) \odot$ represents the choice of which past information to forget and $z_v^t \odot$ represents the choice of remembering the newly generated information.

To summarize, this part constructs each document as a network, represents the words in the document as vertices, and the co-occurring forms between words as edges for the graphical construction of the document, and finally the document is converted into a graphical network and then uses the graph neural network to learn the embedding of the word nodes.

II. C. 4) Readout function

After the word nodes have been sufficiently updated, they are encoded into a graph representation of the document. The final prediction of the model is generated based on this graph representation as follows:

(1) The feature h_v^t of each word in the document is passed through the soft attention layer to obtain the information weight, and then the word feature h_v^t, which is transformed by nonlinear features, is multiplied with the obtained information weight matrix to obtain the graphical vector representation of the document h_v.

(2) A maximum pooling layer is applied by connecting all documents $\left(h_1,\ldots,h_v\right)$ to enable the relational words to have the maximum weight.

(3) Prediction is done by passing the obtained graph vector h_g through a softmax layer and in the final prediction part the cross-entropy function is used to minimize the loss.

The formulae for this part of the computation are reasoned as shown in Eq. (18) to Eq. (21).

$$h_v=\sigma\left(f_1\left(h_v^t\right)\right)\odot\tanh\left(f_2\left(h_v^t\right)\right) \tag{18}$$

$$h_{\mathrm{G}}=\frac{1}{|\mathrm{V}|}\sum_{v\in V}h_v+Maxpooling\left(h_1\ldots h_v\right) \tag{19}$$

$$\hat{y}_{\mathrm{G}}=soft\max\left(Wh_{\mathrm{G}}+b\right) \tag{20}$$

$$\mathrm{L}=-\sum_i y_{\mathrm{G}_i}\log\left(\hat{y}_{\mathrm{G}_i}\right) \tag{21}$$

where f_1 is the soft attention mechanism, while f_2 is the nonlinear feature transformation, W and b are the weights and biases, and $y_{\mathrm{G}i}$ is the i th element in the one-hot label.

II. D.Retrieval model based on topological feature representation of convolutional maps

II. D. 1) Network framework for retrieval models

In recent years, the research of graph neural networks (GNNs) has attracted more and more attention [27]. GCNs are widely used in natural language processing tasks, such as text representation, machine translation, and text classification. In these tasks, the model constructs the graph topology of the text according to the task's object of study sentence or document, and uses convolution in the graph topology to extract the global structured information of the text sequence. In this section we propose a text retrieval model based on graph convolutional topological feature representation, which generates graph convolutional topological feature representation GCTR using graph convolutional network after converting the query and the contextual representation of the document, CLR, into a graph topology. Its network framework is shown in Fig. 3.

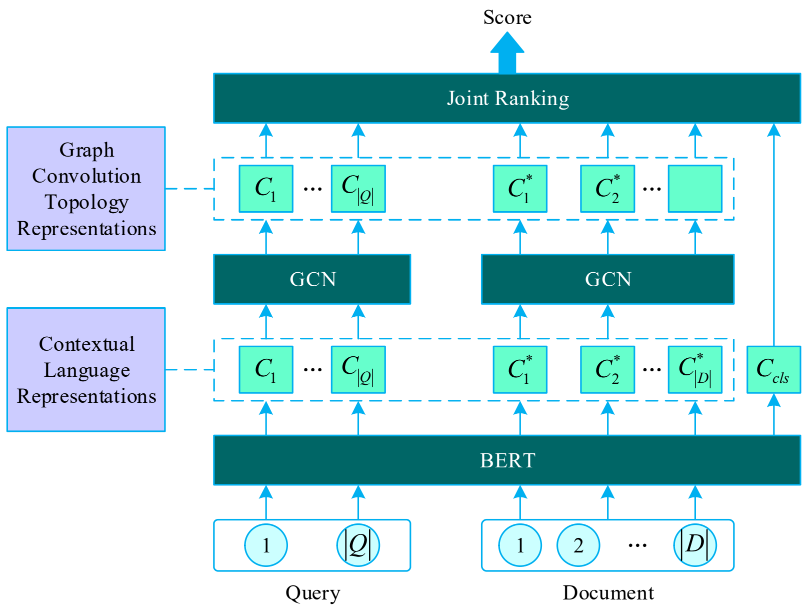

Score
Joint Ranking
Graph Convolution Topology Representations
C_1 … $C_{|Q|}$ C_1^* C_2^* …
GCN
GCN
Contextual Language Representations
C_1 … $C_{|Q|}$ C_1^* C_2^* … $C_{|D|}^*$ C_{cls}
BERT
1 $|Q|$ 1 2 … $|D|$
Query
Document

Figure 3: The retrieval model framework based on the graph of the volume topology

The two steps before text sorting are given in Fig. 3: the pre-training model BERT and the graph convolutional network GCN. Each step produces a new text representation C_i, represented by color boxes of different lengths, the length of the color boxes representing the dimension of the text representation C_i, and each step produces a text representation with a smaller dimension than the dimension of the text representation in the previous step.

II. D. 2) Graph Convolutional Topological Feature Representation

Given the input query Q and document D in Fig. 3, the first step is to generate the contextual representations of the query and document C_q and C_d by pre-training the model BERT in which text words are used as nodes of the graph and the contextual representations of the words are used as feature vectors of the nodes.The input to the GCN is the textual graph Constructed a feature matrix $X \in \mathbb{R}^{n\times m}$, in the case of the document D, in which $n = |D|$, m are the dimensions of the contextual vectors. Meanwhile, we constructed an adjacency matrix S as the adjacency matrix of the graph topology A, A containing the internal compactness information between nodes in the graph topology. The basic structure of a GCN containing two hidden layers is shown in Fig. 4.

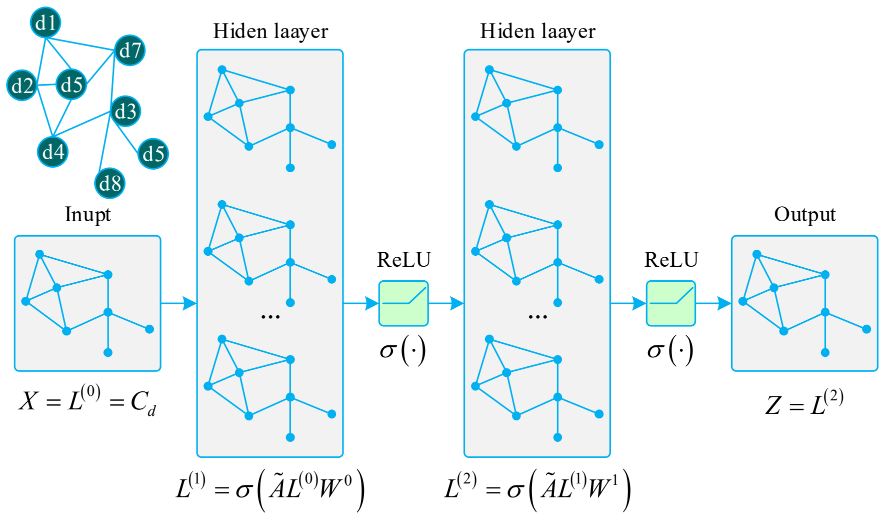


Figure 4: The basic structure of GCN

In the topology diagram G, the relationship between the adjacency matrix $A \in \mathbb{R}^{n\times n}$ and the degree matrix $D \in \mathbb{R}^{n\times n}$ is:

$$D_{ii} = \sum_j A_{ij} \tag{22}$$

Since nodes and themselves are associated in the topology graph, the diagonal elements in the adjacency matrix A need to be all set to 1.

Given that the feature matrices of query Q and document D are $X_q = C_q$ and $X_d = C_d$ respectively, and both $C_q \in \mathbb{R}^{|Q|\times m}$ and $C_d \in \mathbb{R}^{|D|\times m}$ are learned by the pre-trained model BERT. The nodes of the topology graph are the words of the text, and in this paper we choose the cosine distance from node to node as the weights of the connecting edges between nodes. The (edge) weights between node i and node j are calculated as:

$$A_{ij} = \begin{cases} 1 & ifi = j \\ sim(q_i, d_j) & otherwise \end{cases} \tag{23}$$

The similarity $sim(q_i, d_j)$ of node i, j in Eq. is calculated as:

$$sim\left(q_i, d_j\right) = \frac{\left(C_q^i\right)\left(C_d^j\right)^T}{\left\|C_q^i\right\| \cdot \left\|C_d^j\right\|} \tag{24}$$

where C_q^i and C_d^j represent the context vector representation of the i rd keyword in the query and the j th word within the document, respectively.

For the first-order GCN, the output k-dimensional node feature matrix is $L^{(1)} \in \mathbb{R}^{n \times k}$. From the knowledge of graph convolution principle introduced in the previous section, the node feature matrix $L^{(1)}$ is given by:

$$L^{(1)} = \rho\left(\tilde{A}XW_0\right) \tag{25}$$

where $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized adjacency matrix and matrix $W_0 \in \mathbb{R}^{m \times k}$ is the parameter matrix of the GCN. ρ is the activation function, in this paper we choose to use the ReLU function: $\rho(x) = \max(0;x)$.

By stacking multiple GCN layers we can integrate higher order neighborhood information:

$$L^{(j+1)} = \rho\left(\tilde{A}L^jW_j\right) \tag{26}$$

where j represents the strata of the GCN complex and the input of the first-order GCN is $L^0 = X$.

Since then, we define the new graph convolution topology representation for query Q and document D as:

$$\begin{gathered} L_q^{(j+1)} = \rho\left(\tilde{A}_q L_q^j W_q^j\right) \\ L_d^{(j+1)} = \rho\left(\tilde{A}_d L_d^j W_d^j\right) \end{gathered} \tag{27}$$

where $L_q^0 = C_q$, $L_d^0 = C_d$, C_q and C_d represent the context vector representation of the query and document, respectively.

II. D. 3) Convolutional map topological feature representation

Unlike Graph Convolutional Networks GCN, Convolutional Neural Networks CNN mainly deals with one, two or even multi-dimensional data with Euclidean structure.The model structure of CNN consists of convolutional, pooling and fully connected layers. The core idea is to design different sizes of convolutional windows in the convolutional layer to perform convolutional operations on the text sequence, obtain n-grams of feature information within the text sequence, obtain key features through maximum pooling, the pooling layer outputs local feature information of the text sequence, and at the same time, reduces the parameters and computation of the next layer in order to prevent overfitting, and the fully-connected layer maps the features learned in the first two layers into the sample labeling space, to complete the decision-making Classification task. In order to compare the text feature extraction ability of GCN and CNN, in this paper, a layer of CNN is added after the pre-training model BERT, and the BERT outputs the context vector representation.

In order to analyze the performance of GCN and CNN on text feature extraction, this paper adds a layer of CNN in the middle of the pre-training model BERT and GCN layer.CNN outputs a convolutional context representation CCR, and GCN layer outputs a convolutional graph topology feature representation CGTR.CNN generates CCR containing n-grams of local structural information within the text sequence, and GCN generates CGTR containing global structural information of the text. The CNN generates the CCR containing the local structure information of n-grams within the text sequence, and the GCN generates the CGTR containing the global structure information of the text.

Convolutional graph topological feature representation with global structural correlation information. The new text feature representation is used as an input to the deep text ranking model to output the final query-document correlation score.

II. D. 4) Joint sorting

In this chapter, based on the context vector representation output from the pre-training model BERT, a new text feature representation is generated jointly with CNN and GCN, which contains not only text context information, but also local and global structure information. In order to fully utilize the feature information in the text representation, this chapter adopts the same joint ranking model as CEDR: joint pre-training model BERT and deep text ranking models (e.g., PACRR, DRMM, KNRM) for text correlation and ranking.

First, the CLS categorical labeled representation output from the pre-trained model BERT is used to score and rank the query and document relevance by the BERT model. Second, after combining the BERT model with convolutional neural network CNN and graph convolutional network GCN to generate new text feature representations, the deep text ranking model is jointly used to compute the query and document relevance information as an additional ranking estimation. Finally, the CLS vectors output from the BERT model and the relevant feature vectors output from the deep ranking model are connected and mapped to the sample markup space using the fully connected layer to obtain the relevant scoring and ranking of the query and the document as a whole.

III. Experimental design and analysis of results

In this section, two experiments are designed to test the effectiveness of the constructed automatic legal text classification and retrieval model respectively.

III. A. Analysis of the effect of legal text categorization

III. A. 1) Experimental design

In order to more comprehensively verify the effect of FLGNN model in legal text categorization task, this paper constructs two legal fine-grained text categorization datasets for experimental testing.

(1) Datasets

The first dataset, PKULawData, comes from Beida Law Database, and the segmented domain texts of six topics are obtained manually and programmatically, with a total of 5,341 samples, and the statistics of sample categories are shown in Table 1. It can be found that the PKULawData dataset has a large differentiation between categories, and there is a category imbalance problem, in which there are 1,837 samples in the text of buying and selling, only 268 samples in the text of guarantee, and the sample size of the other categories is around 600 to 1,000 samples.

The second dataset, JSCLawData, comes from the internal legal text data of a company in Jiangsu, which contains 17 subject segments of the text, with a total of 3,158 samples, and the statistics of sample categories are shown in Table 2. It can be found that the JSCLawData dataset has a larger number of categories than PKULawData, with less differentiation between categories and smaller sample sizes, while there is a category imbalance problem, in which there are 737 samples of service contracts (C), the number of samples of other categories of texts ranges from about 100 to 250, and the company merger and separation agreement (K), which has the smallest number of samples, has only 31 samples.

In terms of style, the two datasets are both legal texts, but there are also major differences: the PKULawData dataset contains mostly general model texts as basic reference templates, with concise language, no description of the specific details of the terms, and relatively simple content, while the JSCLawData dataset contains mainly example models, which describe in detail the implementation details of the terms of the contract, with more standardized language and more detailed implementation details. The text in the JSCLawData dataset is mainly an example model, which describes in detail the implementation details of each clause of the contract, the language is more standardized and rigorous, and the number of clauses is relatively large and complex.

Referring to the existing research, this paper divides the dataset into training set, validation set and test set according to 8:1:1. The statistical information of the dataset is shown in Table 3, and the average text length of the two datasets is more than 2000, which belongs to long text.

Table 1: Sample Category Statistics of PKULawData Dataset

Categories	Sample size
Vouch	268
Entrustment	721
Construction	773
Service	1087
Buy and sell	1837
Tenancy	611

Table 2: Sample Category Statistics of JSCLawData Dataset

Categories	Sample size
Articles of association	85
Purchase and purchase contract	223
Service contract	737
Intellectual property contract	158
Triad	247
Contract of contract	211
Intermediary and consultative contracts	118
Equity agreement	114
Guarantee contract	99
Labor contract	215
Merger and division agreement	31
Contribution agreement	164
Franchise contract	64
Partnership agreement	132
Lease contract	128
Equity incentive agreement	126
Company rules and regulations	183

Table 3: Summary Statistics of Datasets

Datasets	# Doc	# Train	# Dev	# Test	# Word	# Class	Avg Len
JSCLawData	3158	2488	301	305	42785	19	2379
PKULawData	5341	4275	531	531	55297	8	3127

(2) Text Preprocessing

After processing with the English stop word list (english.stop 571 stop words) and Porter stemmer stem extraction algorithm, 6327 different words were finally obtained. Logarithmic entropy model was used:

$$LogEntropy = (\log_2\left(1+tf_{ij}\right))\bullet(1+\Sigma_j p_{ij}\log_2 p_{ij}/\log_2 n) \tag{28}$$

The weights of the words are calculated.

The distribution of weights after cosine normalization is shown in Fig. 5, where tf_{ij} denotes the word frequency and $p_{ij} = tf_{ij}/\Sigma_j p_{ijo}$.

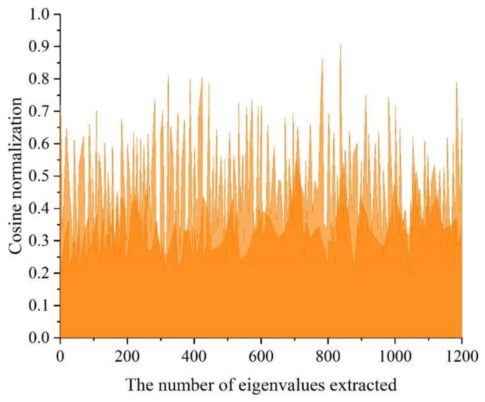


Figure 5: Normalized weight distribution

(3) Baseline model

The FLGNN proposed in this paper is compared experimentally with seven baseline models, FastText, TextCNN, DPCNN, RCNN, BiLSTM_Att, TextGCN, DADGNN and BERT.

All of the above baseline models are deep learning-based neural network models, among which FastText, TextCNN, DPCNN, RCNN, and BiLSTM_Att are sequence-based models; TextGCN and DADGNN are graph-based models; and BERT is based on pre-training.

(4) Experimental Setup

The experiment is run on a cloud server with a single NVIDIA GeForce RTX 3090 (24G) graphics processor, Intel Xeon(R) Gold 6330 CPU, and 80G RAM. The experimental model was implemented using Python 3.9 interpreter, PyTorch deep learning framework, sklearn machine learning library, and DGL graph neural network framework.

For both datasets, this paper uses 12-layer BERT-Base-Chinese as the pre-training model, with the number of model layers set to 5, the feature dimension to 768, epoch to 10, the number of attention heads to 3, and dropout to 0.1, and the Adam optimizer is used for training with a learning rate of 5e-5. On the JSCLawData dataset Batch Size is set to 8, the maximum input sequence length is 512, and the regularization parameter is 1e-9; on the PKULawData dataset Batch size is set to 16, the maximum input sequence length is 512, and the regularization parameter is 1e-4. Each batch is tested on the validation set after training, if the model's loss on the validation set does not decrease after training more than 1,000 batches, the model training is ended and the model parameters with the smallest loss on the validation set are saved.

(5) Evaluation metrics

In this experiment, the performance of the model is evaluated using the macro checking rate P, macro recall rate R, macro $F1$ and accuracy Acc4 metrics, which are calculated as shown in Equation (29)-Equation (32).

$$Macro-P=\frac{1}{L}\sum_{i=1}^{L}\frac{TP_i}{TP_i+FP_i} \tag{29}$$

$$Macro-R=\frac{1}{L}\sum_{i=1}^{L}\frac{TP_i}{TP_i+FN_i} \tag{30}$$

$$Macro-F1=\frac{1}{L}\sum_{i=1}^{L}\frac{2\times P_i\times R_i}{P_i+R_i} \tag{31}$$

$$Acc=\frac{1}{L}\sum_{i=1}^{L}\frac{TP_i+TN_i}{TP_i+TN_i+FP_i+FN_i} \tag{32}$$

where i represents the i rd category of labels in the label set L, TP_i, FP_i, TN_i and FN_i represent the true, false positive, true negative and false negative cases respectively, P_i is the checking accuracy rate of the i th category and R_i is the recall rate of the i th category. The larger the value of the four metrics, the better the experimental results.

III. A. 2) Results and analysis

The DASA-GNN model proposed in this paper is compared with eight baseline models on the JSCLawData dataset and PKULawData dataset using four evaluation metrics (represented by Acc), and the optimal results in each evaluation metric are marked in bold. The experimental results are shown in Table 4.

The experimental results show that the DASA-GNN model achieves the best classification results on both datasets.The Acc of the DASA-GNN model on the JSCLawData dataset is 91.18%, and that on the PKULawData dataset is 95.12%, which is 1.35% and 1.15% higher than that of the baseline model with the optimal classification results, BERT, respectively, which improves the feature mining and feature expression ability of the model.

Table 4: The Experimental Results of Different Models

Model	PKULawData				JSCLawData			
	Acc(%)	P(%)	R(%)	F1(%)	Acc(%)	P(%)	R(%)	F1(%)
FastText	87.21	86.27	88.13	86.98	83.26	86.06	82.97	83.83
TextCNN	93.59	93.6	93.59	93.58	81.87	86.86	81.3	82.18
DPCNN	92.64	92.71	92.64	92.66	80.13	82.36	80.84	79.72
RCNN	93.34	93.27	94.37	93.71	87.08	90.79	86.81	87.41
BiLSTM_Att	93.21	92.9	93.45	93.15	82.91	80.73	79.11	77.69
TextGCN	85.01	86.19	80.49	82.76	84.55	82.77	78.9	80.44

DADGNN	91.68	91.68	90.5	90.92	86.99	90.35	83.83	86.17
BERT	93.97	93.96	93.97	93.96	89.83	90.44	90.06	90.72
DASA-GNN	95.12	94.44	95.86	95.21	91.18	91.51	90.92	91.13

III. A. 3) Ablation experiments

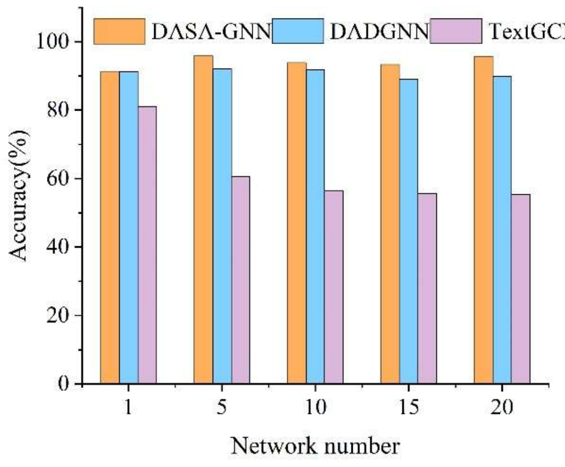
To verify the effectiveness of each module proposed in this paper in DASA-GNN model, three variants of the model are designed and experimented on PKULawData and JSCLawData datasets, and the results of the ablation experiments are shown in Table 5. As can be seen from the table, the Acc of the DASA-GNN model on the JSLawData dataset and PKULawData dataset is 95.12% and 91.18%, respectively, to obtain a more accurate document-level representation.

Table 5: Results of Ablation Experiment

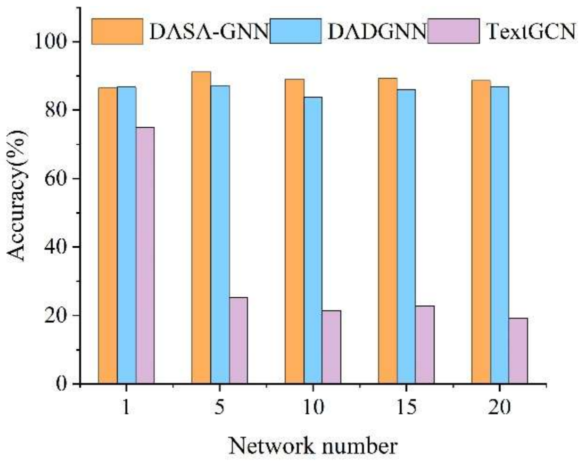
Model	PKULawData				JSCLawData			
	Acc(%)	P(%)	R(%)	F1(%)	Acc(%)	P(%)	R(%)	F1(%)
w/o DADGNN	94.35	94.39	94.29	94.31	90.51	91.33	90.51	90.49
w/o BERT	93.24	93.21	92.32	92.71	86.62	89.78	83.68	85.21
w/o node-level attention	94.16	93.87	94.66	94.23	88.48	89.37	88.54	88.41
DASA-GNN	95.12	94.44	95.86	95.41	91.18	91.51	90.92	91.13

III. A. 4) Sensitivity analysis of the number of network layers

In order to verify whether the DASA-GNN model suffers from over-smoothing problem when the network is stacked in multiple layers. In this paper, we adjusted the number of network layers of the DASA-GNN model, and conducted experiments on two datasets under different conditions of the number of network layers, and used TextGCN and DADGNN in the baseline model as the comparison model. The specific results are shown in Fig. 6. From the experimental results, the classification accuracy of this paper's model remains stable when the number of network layers increases, and the Acc floats only in a small range, and does not undergo a significant decrease with the increase of the number of layers in the model, indicating that the model has a good ability to mitigate over-smoothing.



(a) PKULawData



(b)JSCLawData

Figure 6: Accuracy with Different Numbers of Layers

III. B. Analysis of the effect of legal text retrieval

III. B. 1) Data sets

The dataset used in this paper is LeCaRD, which is the first retrieval dataset of Chinese legal cases, and the cases in this dataset are from the publicly available adjudication documents on the adjudication website. The dataset contains 110 query cases, which are divided into two categories: 77 common query use cases and 30 dispute query use cases. Each query case corresponds to 100 candidate cases. The average text length of query cases is 478 characters, and the average text length of candidate cases is 6347 characters. Different from the previous similarity judgment by supporting cases or related legal knowledge, the similarity judgment of this dataset combines the formulated judgment criteria and the subjectivity evaluation of legal experts, which is more in line with the objective facts.The statistical information of LeCaRD dataset is shown in Table 6.

Table 6: Data set description

Statistics	Quantity
Query case number	110
Candidate case number	10754
Query case text average length	478
Candidate case text average length	6347

III. B. 2) Contrasting models

In order to evaluate the performance and stability of the model in this paper, the model proposed in this paper is compared and experimented with five other different methods, which are the traditional algorithms in the field of information retrieval, BM25 algorithm and LMIR algorithm, the deep learning based KNRM model, the BERT-IR model and the Legal-RoBERTa model.

III. B. 3) Evaluation indicators

According to the evaluation index for the class case retrieval track in the China Assessment of Intelligent Technology in Law (CAIL) competition, this paper adopts NDCG@K, Normalized Discounted Cumulative Gain (NDCG@K) as the evaluation index, and the specific calculation formula is as follows:

$$DCG@K(q)=\sum_{i=1}^{K}\frac{2^{lable_i}-1}{\log(1+i)} \tag{33}$$

$$IDCG@K(q)=IdealDCG@K(q) \tag{34}$$

$$NDCG@K(q)=\frac{DCG@K(q)}{IDCG@K(q)} \tag{35}$$

As shown in Equation (33), DCG@K only cares about the first K sorted retrieval results of query q. If it is related to a relevant document, it will get $2^{lable_i}-1$ a gain, and it will also get $\frac{1}{\log(1+i)}$ a position-related gain or loss depending on the position. Equation (34) represents the maximum value of DCG@K for the first K documents of the ideal case q. In addition, Equation (35) is used to normalize the value of DCG@K. In this paper, NDCG@10, NDCG@20 and NDCG@30 are chosen as evaluation metrics.

III. B. 4) Experimental results and analysis

The experimental results are shown in Table 7, the performance of different models on the same test set varies, in which the model in this paper performs significantly better than several other methods.

In the table, BM25 algorithm, TF-IDF algorithm and LMIR algorithm are three algorithms commonly used in the field of information retrieval. By comparing the experimental results of BM25 algorithm and TF-IDF algorithm, we know that BM25 algorithm has better recall effect than TF-IDF algorithm. Comparing the experimental results of BM25 algorithm and this paper's model, it is found that the effect of this paper's model is better than that of BM25 algorithm, which proves that the fine-ordering module in this paper's model can accurately capture the semantics of the text, and carry out the effective sorting once on the basis of the retrieval results of BM25.

Finally compare the experimental results of the retrieval model combining BM25 algorithm and deep learning algorithm with the model proposed in this paper, the comparison experiment keeps other variables unchanged, and uses different deep learning methods to replace the criminal instrument BERT model in this paper's model for the experiment. As the criminal instrument BERT model is pre-trained on criminal instruments in advance, the ability to obtain the semantics of criminal instruments is stronger than other deep learning methods. Therefore, the model in this paper has better class case retrieval than other deep learning methods.

Table 7: Comparison experiment results

Model	NDCG@10	NDCG@20	NDCG@30
BM25	0.7583	0.7994	0.9011
TF-IDF	0.7221	0.7574	0.8433
LMIR	0.7721	0.8157	0.9037

KNRM	0.6726	0.7144	0.7642
BERT-IR	0.7801	0.8195	0.9053
Legal-RoBERTa	0.789	0.8212	0.8931
BM25+KNRM	0.6911	0.7541	0.7931
BM25+BERT-IR	0.8011	0.8393	0.9114
BM25+Legal-RoBERTa	0.8153	0.8434	0.9181
This model	0.8215	0.8727	0.9307

From the comparison experiments, it can be learned that the fine ranking stage module of this paper's model improves the model retrieval precision, in order to study the impact of the preliminary ranking recall module of this paper's model on the retrieval precision and retrieval efficiency of this paper's model, ablation experiments were designed to study the problem. The experimental design is mainly carried out through the number of candidate cases recalled by BM25 algorithm in the preliminary ranking recall module, and the number of recalled cases is determined as 35, 45, 55, 65 and 75 respectively.The experimental results are shown in Table 8.

Table 8: The case count affects the retrieval results

Recall number	NDCG@10	NDCG@20	NDCG@30	Retrieval time
35	0.7554	0.7904	0.9001	57s
45	0.7899	0.8454	0.9171	67s
55	0.8543	0.8861	0.9261	88s
65	0.8658	0.9033	0.9362	107s
75	0.8679	0.9061	0.9434	141s

The effect of the number of recalled cases on model precision and the effect of the number of recalled cases on model efficiency are shown in Fig. 7 and Fig. 8, respectively. As the number of recalled cases increases, the retrieval precision increases and the retrieval time also increases gradually. Therefore the number of recalls needs to be controlled to achieve an optimal balance between the retrieval precision and retrieval time of the model. From the above figure, it can be concluded that when the number of recall cases is 65, the retrieval precision and efficiency of the model are in the optimal balance. Therefore, the number of recall cases is set to 65 in the initial ranking of this paper.

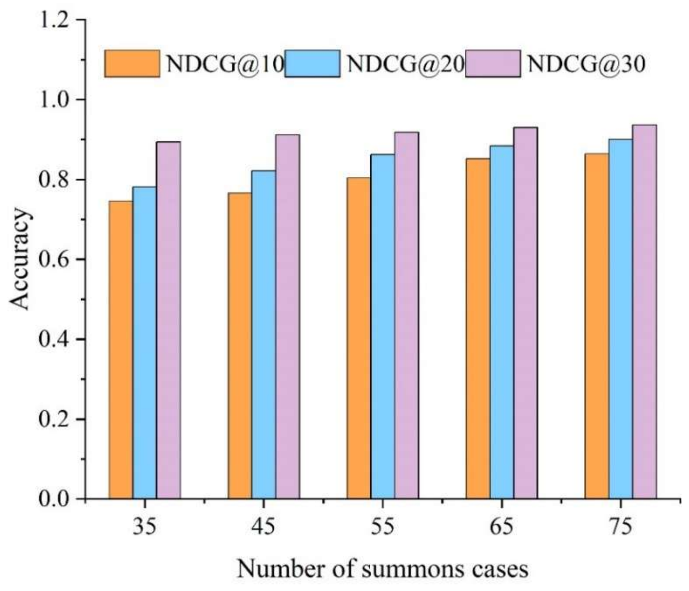


Figure 7: The effect of recall case number on model retrieval accuracy

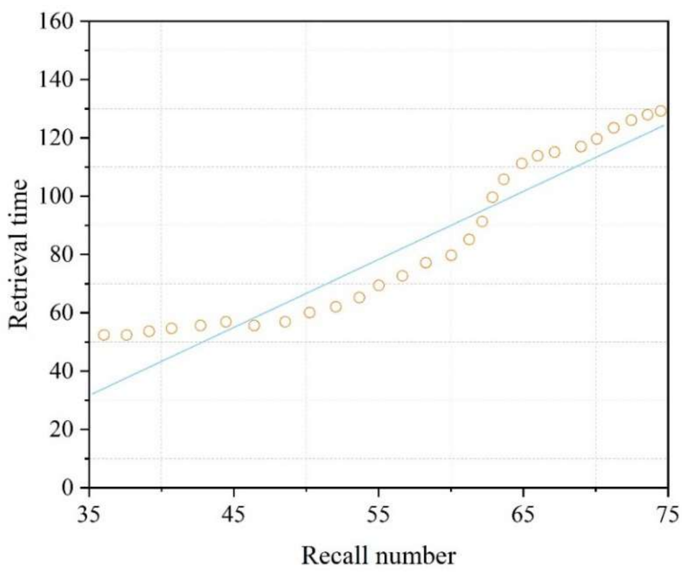


Figure 8: The effect of recall case number on model retrieval efficiency

In order to verify the contribution of the case text screening module to the model retrieval speed, this paper designs the following ablation experiments, respectively, using the residual model with the case text screening module removed for class case retrieval of the test set and using the complete model of this paper for retrieval of the same test set, and the specific results are shown in Table 9.

From the table, it can be learned that because the case text screening module sieves away a large amount of useless information in the candidate cases, the model of this paper greatly shortens the model retrieval time and improves the retrieval efficiency while hardly affecting the model retrieval accuracy.

Table 9: The effect of the text filter module on the retrieval results

Model	NDCG@10	NDCG@20	NDCG@30	Retrieval time
Disability model	0.8731	0.9122	0.9413	155s
Complete model	0.8742	0.9189	0.9426	105s

IV. Conclusion

In this paper, a text categorization model based on DASA-GNN is proposed, which improves the overfitting problem of graph neural networks in text categorization tasks by introducing a self-attention mechanism. Then a text retrieval model based on convolutional graph topological feature representation is proposed to generate new text features using graph convolutional network. The study is carried out through experiments, and the DASA-GNN text classification model has an Acc value of 95.12% on the PKULawData dataset and 91.18% on the JSCLawData dataset from legal contract text, indicating that the DASA-GNN model is able to effectively improve the legal text classification effect. The text retrieval model based on the topological feature representation of convolutional graph is compared and experimented on the legal retrieval dataset LeCaRD, and the model has better retrieval effect with Acc of 95.12% and 91.18% on the JSLawData dataset and PKULawData dataset, respectively.

References

[1] Senyk, S., Churpita, H., Borovska, I., Kucher, T., & Petrovskyi, A. (2022). The problems of defining the legal nature of the court judgement. Amazonia Investiga, 11(56), 48-55.
[2] Liebman, B. L., Roberts, M. E., Stern, R. E., & Wang, A. Z. (2020). Mass digitization of Chinese court decisions: How to use text as data in the field of Chinese law. Journal of Law and Courts, 8(2), 177-201.
[3] Ahl, B., & Sprick, D. (2018). Towards judicial transparency in China: The new public access database for court decisions. China Information, 32(1), 3-22.
[4] Jeandarme, I., Habets, P., & Kennedy, H. (2019). Structured versus unstructured judgment: DUNDRUM-1 compared to court decisions. International journal of law and psychiatry, 64, 205-210.
[5] Tippett, E. C., Alexander, C. S., Branting, K., Morawski, P., Balhana, C., Pfeifer, C., & Bayer, S. (2021). Does lawyering matter? Predicting judicial decisions from legal briefs, and what that means for access to justice. Tex. L. Rev., 100, 1157.
[6] Yi, W. (2023). Study on the Reasoning Traceability and Reform of Family Related Judgment Documents. J. Pol. & L., 16, 55.

[7] Spamann, H., & Klöhn, L. (2016). Justice is less blind, and less legalistic, than we thought: Evidence from an experiment with real judges. The Journal of Legal Studies, 45(2), 255-280.

[8] Zhang, H., Pan, B., & Li, R. (2021). Legal judgment elements extraction approach with law article-aware mechanism. Transactions on Asian and Low-Resource Language Information Processing, 21(3), 1-15.

[9] Zhang, Y., & Huang, X. (2025). 'Human rights' in judicial judgement of China. The International Journal of Human Rights, 1-22.

[10] Guo, J. (2024). A fast retrieval method for legal judgment documents based on multi-granularity semantic interaction understanding. International Journal of Computer Applications in Technology, 74(4), 324-332.

[11] Wagh, R. S., & Anand, D. (2020). Legal document similarity: a multi-criteria decision-making perspective. PeerJ Computer Science, 6, e262.

[12] Ebietomere, E. P., & Ekuobase, G. O. (2019). A Semantic Retrieval System for Case Law. Appl. Comput. Syst., 24(1), 38-48.

[13] Wiggers, G., Verberne, S., Zwenne, G. J., & Van Loon, W. (2022). Exploration of domain relevance by legal professionals in information retrieval systems. Legal Information Management, 22(1), 49-67.

[14] Solihin, F., Budi, I., Aji, R. F., & Makarim, E. (2021). Advancement of information extraction use in legal documents. International Review of Law, Computers & Technology, 35(3), 322-351.

[15] Zadgaonkar, A. V., & Agrawal, A. J. (2021). An overview of information extraction techniques for legal document analysis and processing. International Journal of Electrical & Computer Engineering (2088-8708), 11(6).

[16] Donnelly, J., & Roegiest, A. (2020, October). The utility of context when extracting entities from legal documents. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (pp. 2397-2404).

[17] Shivananda, S., Mathews, L. M., Nikkam, S., Kambli, P., Ingale, R. D., & Koparde, P. (2024, November). Revolutionizing Legal Case Analysis through Advanced Event Extraction. In 2024 International Conference on Recent Advances in Science and Engineering Technology (ICRASET) (pp. 1-6). IEEE.

[18] Adhikary, S., Sen, P., Roy, D., & Ghosh, K. (2024). A case study for automated attribute extraction from legal documents using large language models. Artificial Intelligence and Law, 1-22.

[19] Satterfield, N., Holbrooka, P., & Wilcoxa, T. (2024). Fine-tuning llama with case law data to improve legal domain performance. OSF Preprints.

[20] Buey, M. G., Roman, C., Garrido, A. L., Bobed, C., & Mena, E. (2019). Automatic legal document analysis: Improving the results of information extraction processes using an ontology. Intelligent Methods and Big Data in Industrial Applications, 333-351.

[21] Resck, L. E., Ponciano, J. R., Nonato, L. G., & Poco, J. (2022). LegalVis: Exploring and inferring precedent citations in legal documents. IEEE Transactions on Visualization and Computer Graphics, 29(6), 3105-3120.

[22] Silva, H., António, N., & Bacao, F. (2022, August). A rapid semi-automated literature review on legal precedents retrieval. In EPIA Conference on Artificial Intelligence (pp. 53-65). Cham: Springer International Publishing.

[23] Correia, F. A., Almeida, A. A., Nunes, J. L., Santos, K. G., Hartmann, I. A., Silva, F. A., & Lopes, H. (2022). Fine-grained legal entity annotation: A case study on the Brazilian Supreme Court. Information Processing & Management, 59(1), 102794.

[24] Mentzingen, H., António, N., & Bacao, F. (2025). Effectiveness in retrieving legal precedents: exploring text summarization and cutting-edge language models toward a cost-efficient approach. Artificial Intelligence and Law, 1-21.

[25] Ribeiro de Faria, J., Xie, H., & Steffek, F. (2025). Information extraction from employment tribunal judgments using a large language model. Artificial Intelligence and Law, 1-22.

[26] Yan Jiang,Beilong Luo,Yuan Jiang,Min Liu,Shuoyu Liu & Liuliu Peng. (2025). Prediction of long-period ground motion responses for high-rise buildings using physics-assisted fully convolutional neural network. Journal of Building Engineering,104,112264-112264.

[27] Jie Chang,Haodong Ren,Zuoyong Li,Yinlong Xu & Taotao Lai. (2025). A unified transductive and inductive learning framework for Few-Shot Learning using Graph Neural Networks. Applied Soft Computing,173,112928-112928.