IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 April 2024, accepted 7 June 2024, date of publication 26 June 2024, date of current version 15 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3419139

RESEARCH ARTICLE

A Heterogeneous Graph Based on Legal Documents and Legal Statute Hierarchy for Chinese Legal Case Retrieval

MENGZHE HEI, QINGBAO LIU, SHENG ZHANG, HONGLIN SHI, JIASHUN DUAN, AND XIN ZHANG

College of Systems Engineering, National University of Defense Technology, Changsha 410000, China

Corresponding author: Xin Zhang (shinezhang_nudt@163.com)

This work was supported in part by the National Ministries and Commissions Foundation of China under Grant 41412060502; and in part by the National Natural Science Foundation Youth Program, China, under Grant 62102431.

ABSTRACT Legal case retrieval, which aims at retrieving similar legal cases based on the given query cases, plays an important role in intelligent legal systems. Existing methods including text-based and network-based methods have achieved notable advancements. However, text-based methods tend to overlook the significance of legal statutes and network-based information. Network-based methods ignore the rich text-based information in legal documents and are not applicable when legal citation networks are sparse. Moreover, most of the existing methods cannot be applied to inductive situations where new query cases continue to emerge. To overcome these issues, we propose a heterogeneous graph based on legal documents and legal statute hierarchy called LeDSGra, and utilize a heterogeneous graph representation learning method to combine both text-based and network-based information. Additionally, we design a module to identify relevant legal articles for query cases to connect the query cases to the existing citation network whereby the LeDSGra can be applied in inductive situations. Extensive experiments have been conducted on two real-world legal datasets which are LeCaRD and ELAM, and the experimental results demonstrate that our proposed method significantly improve the performance on the task of legal case retrieval. Moreover, our model outperforms other strong baseline models on both two datasets.

INDEX TERMS Legal case retrieval, heterogeneous graph, legal documents similarity.

I. INTRODUCTION

Recently, the application of AI technology in the legal field has attracted the attention of many researchers. Relevant studies mainly include legal judgment prediction [1], [2], legal case match [3], [4], legal statute identification [5], [6], legal case clustering [7], [8] and legal case retrieval [4], [9]. The task of legal case retrieval is aimed to identify legal cases that are similar to the description of a given query case. Similar legal case retrieval can provide additional support for the judgment of the target case and thus have an effect on the fairness of legal decisions. Given the substantial volume of prior cases, it is inefficient for decision-makers to manually identify similar legal case documents. Hence, there is a pressing need for automated tools to retrieve similar cases. To this end, similar legal case retrieval system which can conveniently access relevant case documents can increase the effectiveness of judiciary.

A key task for the similar legal case retrieval system is to measure the similarity of two documents, and this task has been studied for many years. The existing methods can be broadly classified into text-based and network-based methods [10]. The text-based methods rely on the textual contents of the documents. Some early approaches were based on rules-based and statistical methods [11], [12], [13]. For example, Kumar et al. [13] regarded documents as a collection of word and measured the similarity by calculating Term Frequency-Inverse Document

The associate editor coordinating the review of this manuscript and approving it for publication was Yiqi Liu.

Frequency (TF-IDF) scores of the documents. Then researchers began to use learning-based measures, which typically generate representations of each legal case document and measure the similarity through the representations. For example, Huang et al. [14] and Shen et al. [15] encoded two documents with a fully connected network, convolutional neural network, or recurrent neural network. Currently, pretrained language models are widely used. For example, Hong et al. [16] employed BERT model [17] to obtain the representations of legal cases and measure the similarity. The network-based methods rely on the information from network consisting of legal cases documents and other law legal entities. Some network-based approaches construct a Precedent Citation Network (PCNet) which considers citations to prior cases [13], [18]. Apart from the citation network consisting of prior cases, Bhattacharya et al. [19] also considered the hierarchy of legal statutes and took it into the construction of the network.

Although methods above have achieved breakthrough progress, the task to retrieve similar cases still have the following challenges. Firstly, the topological relationships between different legal entities play an important role in legal case retrieval, but typical text-based methods merely consider legal case documents and tend to overlook the significance of network-based information. Secondly, network-based methods ignore the rich text-based information in the legal documents and are not meaningfully applicable when legal citation networks are sparse. Thirdly, in real-world scenarios where new query cases continue to emerge, query cases are not present in the training data. In such inductive situations, many methods are either unavailable or less effective.

To overcome the challenges above, we propose a heterogeneous graph based on legal documents and legal statute hierarchy called LeDSGra which is shown in the Fig.1. Hierarchy of legal statutes is a collection of acts, parts, chapters, sections and articles. In the LeDSGra, the vertices represent legal entities including the hierarchical structures above and legal case documents, the edges represent the relationships between them. Therefore, the LeDSGra contains textual contents and topological relationships of legal statute hierarchy and legal case documents. Moreover, we utilize a metapath-based heterogeneous graph embedding learning method to combine the text-based and network-based information from the LeDSGra. Besides, to enhance the performance in inductive situations, we first train a model to identify applicable legal laws for query cases and connect new legal query cases with the existing graph. In this way, the metapath based method can generate appropriate representations for query cases.

Specifically, we define 4 types of metapath which accord with the legal common sense and similar patterns in different scenarios. Then we predict the relevant articles of law for the query case through a legal law identification model. We take charge information, which can provide explicit knowledge to identify relevant legal articles, into consideration and jointly model charge prediction and legal article identification in a unified framework. Finally, we use a metapath-based embedding method to generate representations of legal cases and measure the similarity. The embedding method we propose includes three components which are node representation initialization, intra-metapath aggregation, and inter-metapath aggregation. Node representation initialization employ a pretrain language model to generate the initial embeddings of nodes in the LeDSGra and project them into the same latent space. During the process of intra-metapath aggregation for every metapath, we employ a relational rotation method to model every edges in each metapath instance to obtain the representation of instance and fuse the information from all instances using a attention mechanism. In this way, our method captures the text-based and network-based information of every metapath from both nodes and edges in metapath instances. At last, we conduct inter-metapath aggregation which rely on the attention mechanism to fuse the representations obtained from multiple metapaths into final representation of the target node in the LeDSGra. Through combining multiple metapaths, our model can capture rich structural and textual information stored in the LeDSGra and generate node representations for every legal cases. At last, we can measure the similarity through node representations.

Our main contributions are listed as follows:

- We propose a heterogeneous graph called LeDSGra and a heterogeneous graph representation learning method, which can combine text-based and network-based signals for legal case retrieval. To our best knowledge, we are the first to combine the structural and textual information of legal documents and hierarchy of legal statutes in Chinese judiciary.
- We design a module to identify relevant legal articles of law for query cases, which incorporates the corresponding isolated nodes into the existing heterogeneous graph. This enables the performance of the heterogeneous graph representation learning algorithm in inductive situations, thereby improving the performance of our model in real-world scenarios.
- We introduce the hierarchy of legal statutes in Chinese judiciary to two real-world legal case retrieval datasets LeCaRD [20] and ELAM [21]. Then we conduct extensive experiments to evaluate the performance of our method. The results show that our model achieves state-of-the-art performance in comparison to other strong baselines.

The remainder of this paper is organized as follows. Section II presents related works, including research advances in text-based, network-based and combining methods. Section III gives a clear description of the LeDSGra and metapaths used in this work as well as formal definitions of involved terminologies. Section IV formally describes our task. Section V first introduces the overall structure of our proposed model and then details each component. Section VI describes the experimental settings. Section VII presents the experimental results and provides a detailed analysis from

multiple perspectives. Section VIII concludes our work and looks forward to future work.

II. RELATED WORK

In this work, we mainly consider the task of similar legal case retrieval, whose key technique is measuring the similarity of two legal case documents. The method we propose combine two types of approaches which are text-based methods and network-based methods. Therefore, we will introduce the related works from three aspects: (i) text-based methods, (ii) network-based methods, and (iii) combining methods.

A. TEXT-BASED METHODS

Text-based methods rely on the textual contents of the legal case documents. At early stage, some researchers measured the similarity based on string comparisons. Cohen et al. [12] measured the similarity of two documents by matching the text strings of them. Kumar et al. [13] measured the similarity of two documents using all-term cosine similarity and legal-term cosine similarity. Both two measures were relied on the TF-IDF scores of terms in the case documents. This way, case documents are represented as vectors, where each component correspond to the score of a term, and when the vectors of two documents are generated, the cosine similarity are calculated to measure the similarity. These approaches rely solely on the frequency of specific words or terms, rendering them inadequate in scenarios where two similar case documents share only a limited vocabulary overlap.

To overcome the challenge above, some researchers represented the documents as low dimension, continuous vectors to learn the semantics information from the textual contents. Mandal et al. [22] used multiple methods including topic models, word embeddings and document embeddings to gain the representations of the target documents, and these methods of document summary were proposed to filter out the redundant parts of long-form legal case documents. In recent times, deep learning methods, which have gained widespread popularity in the field of natural language processing, are also employed to measure the similarity of legal case documents. Hong et al. [16] employed BERT model as the encoding layer to capture long-range dependencies in the case documents. Shao et al. [23] applied BERT model to generate representations of each paragraph of legal cases, followed by the use of a recurrent neural network (RNN) to predict their similarity. Bhattacharya et al. [24] generated the representations of each paragraph of the documents, and compared the similarity of them in paragraph level. Xiao et al. [25] proposed a pre-trained legal language model using a large number of criminal and civil case documents and used the model on a variety of LegalAI tasks.

Traditional text-based methods merely consider the textual contents of legal case documents, ignore the significance of legal statutes and network-based information in the citation network which are rich in legal-domain knowledge. Therefore, it is beneficial to introduce structural knowledge to text-based methods.

B. NETWORK-BASED

The network-based usually construct a Precedent Citation Network (PCNet), in which the nodes are legal cases documents and edges denote the citations of the cases, and based on the PCNet documents are judged whether they are similar. Krumar et al. [13] inferred the similarity of documents by calculating the Jaccard similarity index between sets of out-citations and in-citations from document clusters, which were called Bibliographic Coupling and Co-citation. Minocha et al. [18] determined whether the sets of precedent citations (out-citation) occur in the same cluster to measure the similarity of two documents. Liu [26] improved the Bibliographic Coupling through employing the titles of the out-citation references which provided the model with more information. Bhattacharya et al. [24] utilized Node2Vec [27] to map legal cases into vector embeddings and evaluated similarity based on these embeddings. In addition to legal case documents, Bhattacharya et al. [19] recognized the importance role of the legal statute hierarchy and incorporated it into the heterogeneous graph.

Network-based methods only consider the structural information of the constructed network and overlook the significant role of text-based information stored in the textual contents of legal documents. Moreover, network-based methods are not meaningfully applicable when the network is sparse [28]. Therefore, combining both text-based and network-based information may improve the accuracy of similarity case documents measurement.

C. COMBINING METHODS

There are no general model specifically designs to combine textual contents and network information. Therefore, methods that try to fuse both two types of information for the task of legal case retrieval develop separately. Existing methods usually construct a specific graph first, and then utilize the representations of the documents generated by graph representation learning algorithms to measure the similarity. Bhattacharya et al. [4] made an attempt to combine textual contents and citation network in legal documents for similarity measurement. In their work, they utilized TADW, graph convolutional network (GCN) and GraphSage to fuse the information from textual contents and network. Bi et al. [10] built a legal heterogeneous graph consisting of legal documents and legal entries. Based on the heterogeneous graph, they employed a sample strategy to aggregate information from neighbor nodes of target node and generated vector representations of it. In this work, legal entities which refer to a legal-related entry in the encyclopedia were utilized to enhance document representations as legal-domain knowledge. Tang et al. [29] also utilized structural information by means of building a text-attributed case graph and an edge graph attention layer.

Although methods mentioned above have achieved great success in combining two types of information to generate better representations of documents, they still lack specific improvements for the task of legal case retrieval. As a result, they are unable to effectively leverage the two types of information in the legal domain.

III. CONSTRUCTION OF THE LEDSGRA AND PRELIMINARY

In this section, we first give a clear description of the LeDSGra (heterogeneous graph based on legal documents and legal statute hierarchy). Then we introduce the metapaths used in this work and semantics of them. Finally, formal definitions of involved terminologies are given.

A. CONSTRUCTION OF THE LEDSGRA

LeDSGra, whose structure is shown in the Fig.1, comprises textual contents and topological relationships of legal statutes and legal case documents. In Chinese judiciary, legal case documents include basic case information, court analysis and judgment of prior cases. The hierarchy of legal statutes is a collection of acts, parts, chapters, sections, and articles. Acts can be divided into multiple parts, each part can be further divided into chapters, and chapters and sections can be further subdivided in the similar manner, but not all hierarchical structures exist in a single act. Therefore, there are 6 types of nodes in the LeDSGra which are legal case documents, acts, parts, chapters, sections and articles. There are 3 types of edges – hierarchy edges, citation edges and similarity edges. The three types of edges are described below.

- Hierarchy edges (*hierarchical structure* $\rightarrow$ *hierarchical structure*): this type of edge (shown as solid black arrows in Fig.1) indicates hierarchical relationships within different and contiguous node structures (acts/parts/chapters/sections/articles). When an act is connected to a part, it signifies that the part is a constituent of the act. For example, $section_1$ is connected to $article_1$ in the Fig.1, which means that $section_1$ is a part of $article_1$.
- Citation edges (*document*$-$*article*): this type of edge (shown as solid blue lines in Fig.1) indicates an article of law is applicable to a legal case document or a legal case document cites to an article of law. E.g. legal case document d_1 cites $article_1$ in the Fig.1.
- Similarity edges: similarity edges have two different types. (1) *document* $-$ *document*: this type of edge (shown as blue dotted lines in Fig.1) implies two legal case documents are similar. In the Chinese judiciary system, case documents do not typically include citations to prior cases. Therefore, in LeDSGra, we define that if two documents are connected, it indicates that they are similar case documents. E.g. document d_1 is similar to document d_2 in the Fig.1. (2) *article* $-$ *article*: this type of edge (shown as black dotted lines in Fig.1) occurs when two articles of law from different acts have similar provisions or can be applied in similar scenarios. E.g. $article_1$ is similar to $article_2$.

To utilize both structural and textual information from the LeDSGra, textual contents of various nodes serve as the input of the language model to generate initial vector embeddings. For the hierarchy of legal statute, we utilize all the textual context of hierarchical structures to obtain the vector embeddings. As for the legal case documents, we solely leverage the basic case information of they to avoid exceeding the capabilities of the language model.

B. METAPATHS

Based on real-world legal scenarios, cases that cite the same or similar legal articles are more likely to be similar. Additionally, if both cases are similar to the same case, then it is highly likely that these two cases are also similar. In order to capture the situations above, we define 4 different metapaths which comply with the legal common sense and similarity patterns. With these metapaths specifically designed for case similarity, indications of similarity between legal case documents can be deduced. The metapaths we define are as follow:

- P_1 **document-article-document**: when two documents cite the same article of law. E.g., legal case documents d_1 and d_4 both cite $article_1$ in Fig.1.
- P_2 **document-article-article-document**: when two case documents cite two similar articles of law respectively. E.g., legal case documents d_1 cites $article_1$,

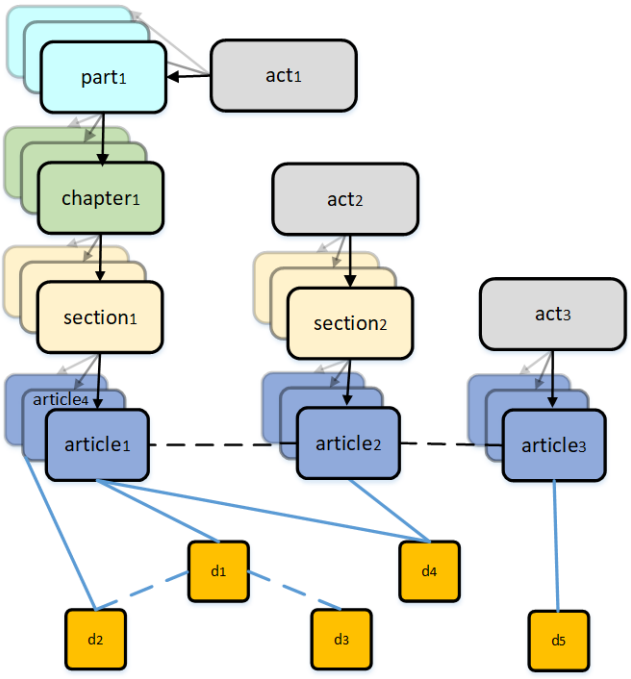

FIGURE 1. An illustration of the LeDSGra structure. The grey, blue, green, yellow, purple and orange boxes represent acts, parts, chapters, 1sections, articles and legal case documents respectively. There are three acts in the figure, act_1 have all levels of the hierarchy, but act_2 only have levels of section and article, act_3 only have a level of article.

d_4 cites $article_2$, and $article_1$ are similar to $article_2$ in Fig.1.

- P_3 **document-article-section-article-document**: when two case documents cite two different articles from the same section. E.g., legal case documents d_1 cites $article_1$, d_2 cites $article_4$, and $article_1$ and $article_4$ are both from $section_1$ in Fig.1.
- P_4 **document-document-document**: when two case documents are connected to the same case document. E.g., in Fig.1, d_1 is connected to d_2, and d_1 is connected to d_3.

These metapaths may only capture partial legal scenarios that reflect the similarity between legal case documents. In the future, more metapaths for legal domains can be defined to further enhance the performance of the model. Additionally, if we transform the similarity relationship among case documents in the defined metapaths into the relationship of citation, our method can also be applied to the common legal system.

C. PRELIMINARY

In order to clearly describe our method, we give formal definitions of some important terminologies of heterogeneous graphs and task of heterogeneous graph representation learning in LeDSGra.

- **Heterogeneous graph.** A heterogeneous graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and sets of node types and edge types $\mathcal{A}$, $\mathcal{R}$ respectively, with $|\mathcal{A}| + |\mathcal{R}| > 2$.
- **Metapath.** A metapath P is defined as a sequence schema of node types and edge types in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$ where $A_1, A_2, \ldots, A_{l+1}$ and $R_1, R_2, \ldots, R_l$ denote the types of nodes and types of edges in the heterogeneous graph but not instances of nodes and edges.
- **Metapath Instance.** Given a metapath P of a heterogeneous graph, a metapath instance p of P is defined as a node sequence in the graph whose node types and edge types fully follow the schema defined by P.
- **Metapath-based Neighbor.** Given a metapath P of a heterogeneous graph, the metapath-based neighbors $\mathcal{N}_v^P$ of a node v is defined as the set of nodes that connect with node v via all the metapath instances of P. Moreover, if one node is connected by node v by two different instances, it is regarded as two different nodes in $\mathcal{N}_v^P$.

For example, for the metapath $document - article - document$ in the Fig.1, $d_1 - article_1 - d_4$ is a metapath instance of it, and d_4 is a metapath-based neighbor of d_1.

IV. TASK FORMULATION

This work primarily focuses on retrieving similar legal cases for the new case, which involves two tasks: legal article identification and legal case retrieval. Formally, the fact description of the new case $\mathbf{t}_{new}$ can be seen as a word sequence $\mathbf{t}_{new} = \{t_1, \ldots, t_n\}$, where n represents the sequence length, $t_i \in T$, and T is a fixed vocabulary. Let $A = \{a_1, a_2, \ldots, a_{|A|}\}$ denotes the set of all articles of law. Since the structural features of the new case is not visible to the model, we first train a function $\mathcal{G}(\cdot)$ such that $\mathcal{G}(\mathbf{t}_{new}, A) = \hat{y}_1$, where $\hat{y}_1 = \{0, 1\}^{|A|}$ and $\hat{y}_1[s] \in \{0, 1\}$ denoting whether $a_s (s = 1, 2, \ldots, |A|)$ is predicted to be relevant to the new case.

Since the LeDSGra is a heterogeneous graph, retrieving similar legal case documents for the new case can be regarded as a link prediction task. Let v_{new} denotes the node of the new case, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes the LeDSGra, where $\mathcal{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|-1}, v_{new}\}$. Our task is to learn a function $\mathcal{F}(\cdot)$ such that $\mathcal{F}(\mathcal{G}) = \hat{y}_2$, where $\hat{y}_2 = \{0, 1\}^{|\mathcal{V}|-1}$ and $\hat{y}_2[m] \in \{0, 1\}$ denoting whether $v_m (m = 1, 2, \ldots, |\mathcal{G}| - 1)$ is predicted to be connected with v_{new}.

V. METHODOLOGY

In this section, we present a comprehensive introduction to legal article identification and legal case retrieval in LeDSGra. As shown in the Fig. 2, legal article identification serves as an auxiliary task to generate relevant legal articles for the new case so that the node representing the new case is connected to the LeDSGra. Subsequently, link prediction task is conduct based on the LeDSGra and final results are obtained.

A. LEGAL ARTICLE IDENTIFICATION

As aforementioned, the LeDSGra is unable to perceive the structural features of new query cases. Thus, new query cases exist as isolated nodes in the LeDSGra, and the model cannot generate effective vector representations for them. To address this issue, we first predict relevant legal articles for new cases in order to establish connections between the new cases and the heterogeneous graph.

However, the abundance of candidate legal articles poses challenges to the task. Therefore, we introduce charge information into consideration and jointly model charge prediction and legal article identification in a unified framework. The reason for introducing charge-aware information is that it can provide explicit knowledge to identify relevant legal articles.

As illustrated in Fig.3, the BERT model converts each word $t_i \in \mathbf{t}_{new}$ into token-level embedding, we have:

$$\begin{aligned} \mathbf{x}_{new} &= \text{BERT}(\mathbf{t}_{new}) \\ &= [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]. \end{aligned} \tag{1}$$

We employ an attention mechanism to select relevant information from facts and get the results of charge prediction:

$$\begin{aligned} \alpha_i &= \text{softmax}\left(\tanh(\mathbf{a}^\top \cdot \mathbf{x}_i)\right) \\ \mathbf{h}_c &= \sum_{i=1}^{n} \alpha_i \cdot \mathbf{x}_i \\ \hat{\mathbf{y}}_c &= \text{softmax}(\mathbf{W}_c \cdot \mathbf{h}_c + \mathbf{b}_c), \end{aligned} \tag{2}$$

where α_i indicates the weight of word-level embedding, $\mathbf{a}$ is the parameterized context vector, $\mathbf{h}_c$ and and $\hat{\mathbf{y}}_c$ are final representations and results of charge prediction.

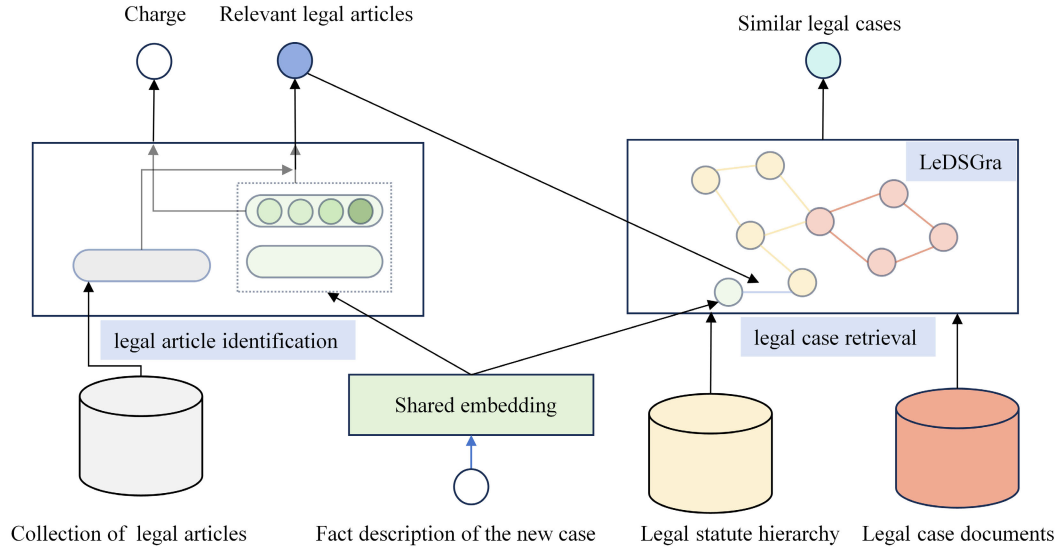

FIGURE 2. The framework of our proposed model. The legal article identification model on the left is an auxiliary task, where the generated relevant legal articles for the new case enable the corresponding nodes in LeDSGra to be connected to the existing heterogeneous graph. The main of similar case retrieval On the right is the main task, which utilizes legal documents and legal statute hierarchy to construct LeDSGra. Additionally, it employs heterogeneous graph representation learning methods to perform link prediction tasks and obtain the final results.

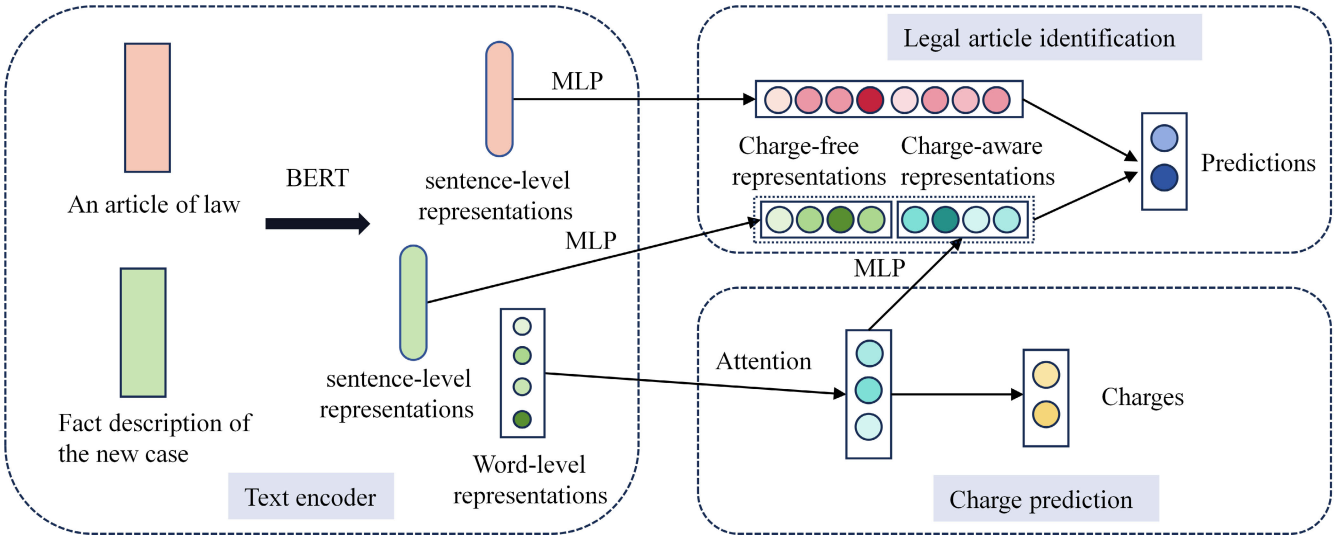

FIGURE 3. An illustration of model for legal article identification. Charge prediction and legal article identification are jointly modeled in the form of multitask learning.

To get charge-free information, sentence-level representations generated by the BERT model is utilized:

$$\begin{aligned} \mathbf{h}_{new} &= \text{BERT}(\mathbf{t}_{new}) \\ \mathbf{h}_f &= \mathbf{W}_f \cdot \mathbf{h}_{new} + \mathbf{b}_f, \end{aligned} \tag{3}$$

where $\mathbf{h}_{new}$ are the sentence-level representations of the new case, $\mathbf{h}_f \in \mathbb{R}^d$ are the charge-free representations, $\mathbf{W}_f$ and $\mathbf{b}_f$ are parametric weight matrices specific for charge-free information.

When generating charge-aware representations, it is important to project the $\mathbf{h}_c$ onto the same latent space as charge-free representations:

$$\mathbf{h}_a = \mathbf{W}_a \cdot \mathbf{h}_c + \mathbf{b}_a, \tag{4}$$

where $\mathbf{h}_a \in \mathbb{R}^d$ are the charge-aware representations, $\mathbf{W}_a$ and $\mathbf{b}_a$ are parametric weight matrices specific for charge-aware information.

To integrate the charge-aware and charge-free information, the two representations are concatenated into the final representations $\mathbf{h}_{case} \in \mathbb{R}^{2d}$:

$$\mathbf{h}_{case} = \mathbf{h}_a \oplus \mathbf{h}_f, \tag{5}$$

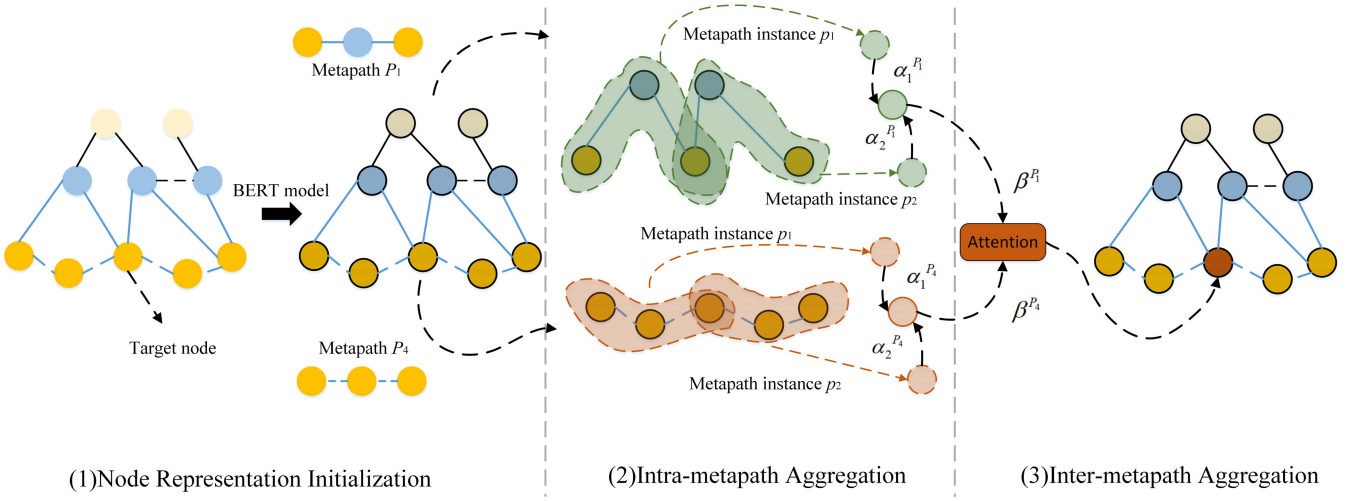

FIGURE 4. An illustration of the heterogeneous graph representations learning in LeDSGra. The learning method consists of three components, where are node representation initialization, intra-metapath and inter-metapath aggregation. In the first step, we employ BERT model to acquire vector representations for each legal document, subsequently mapping them into the same vector space. Moving to the second step, we selectively exemplify the first and fourth metapaths with two instances for each metapath. We obtain representations for each metapath through a weighted aggregation of the respective instances. At last, the representations of the target node is generated through a weighted aggregation of each metapath with attention mechanism.

then we generate the sentence-level representations of each article of law with the BERT model:

$$\mathbf{h}_{law} = \text{BERT}(a_j), \tag{6}$$

where $\mathbf{h}_{law} \in \mathbb{R}^{2d}$ and $a_j (j = 1, 2, \ldots, |A|)$ is an article of law. The cosine distance is employed to measure the relevance between the new legal case and each legal article of law:

$$\begin{aligned} \hat{y}_j &= \cos(\mathbf{h}_{case}, \mathbf{h}_{law}) \\ \hat{\mathbf{y}} &= [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|A|}]. \end{aligned} \tag{7}$$

After legal law identification for a new case, LeDSGra is augmented with a new node and some edges.

B. LEGAL CASE RETRIEVAL

Legal case retrieval aims to predict similar legal case of the query cases. As shown in the Fig.4, it includes node representation initialization, information aggregation based on metapaths.

1) NODE REPRESENTATION INITIALIZATION

In the LeDSGra, the initial attributes of nodes are vectors representations of their textual contents generated by the pre-trained language model. Given the breakthrough progress achieved by the BERT model [17], we use the BERT model to obtain the initial attributes of each note, we have

$$\mathbf{x}_v^A = \text{BERT}(t_v^A), \tag{8}$$

where t_v^A denotes the textual content of the node $v \in \mathcal{V}_A$ of type of A, $\mathbf{x}_v^A$ denotes the vector representations of node v.

The vector representations of nodes are derived from various corpora, such as basic case information of case documents and contents of legal articles. Therefore, the vector representations may lie in the different latent spaces, which is detrimental to generating target node representations by aggregating these representations. In this case, we should project the vector representations from different types of node to the same latent space. For a node v of type A, we have

$$\mathbf{h}'_v = \mathrm{W}_A \cdot \mathbf{x}_v^A \tag{9}$$

where $\mathbf{x}_v^A \in \mathbb{R}^{d_A}$ denotes the vector representations of node v generated from the BERT model, $\mathbf{h}'_v \in \mathbb{R}^{d'}$ denotes the projected vector representations for node v, $\mathrm{W}_A \in \mathbb{R}^{d' \times d_A}$ is the parametric weight matrix specific for nodes of type A.

Using BERT model, the textual contents from different legal text resources is transformed into vector representations that can be calculated by the computer. Also, node representation transformation projects nodes of different types to the same latent space, which facilitates the aggregation process based on node features.

2) INTRA-METAPATH AGGREGATION

Given a target node v and a metapath P, the intra-metapath aggregation layers are aimed to learn the structural and textual information stored in the metapath instances starting with the target node v.

While the textual information of the nodes is extracted using pre-trained language models, it is crucial to focus on modeling the edges in order to better uncover the topological relationships between different legal entities in various legal situation represented by metapaths. Inspired by Fu et al. [30] and Sun et al. [31], we use relational rotation to model nodes and edges in the metapath instances, which can take all nodes and edges into consideration rather than ignore them like most network based methods. Then the attention mechanism

is employed to fuse the information from all the metapath instances of metapath P.

Given a metapath instance $P(v,u) = (n_0, n_1, \ldots, n_M)$, where $u \in \mathcal{N}_v^P$ is a metapath-based neighbor of v, $n_0 = u$ and $n_M = v$, the metapath instance encoder is formulated as:

$$\begin{aligned} \mathbf{q}_0 &= \mathbf{h}'_{n_0} = \mathbf{h}'_u, \\ \mathbf{q}_i &= \mathbf{h}'_{n_i} + \mathbf{q}_{i-1} \odot \mathbf{r}_i, \\ \mathbf{h}_{P(v,u)} &= \frac{\mathbf{q}_n}{n+1}, \end{aligned} \tag{10}$$

where $\odot$ is the element-wise product, $r_i \in \mathbb{R}^{d'}$ is a learnable parametric vector for the edge R_i, $\mathbf{h}_{P(v,u)}$ is the vector representation of $P(v,u)$. By introducing a learnable parametric vector for the relations between two nodes, the model can acquire the latent semantics of the edges, and thus a better vector representation of the target node can be generated.

Now we obtain the vector representations of metapath instances of P, we attentively fuse these representations into the vector representation of the target node by adopting the attention mechanism proposed by literature [32]. The key point of this method is metapath instances contribute to the target representation differently according to their importance, which can be modeled by a learned weight factor α_{vu} for each instance. We have:

$$\begin{aligned} e_{vu}^P &= \text{LeakyReLU}\left(\mathbf{a}_P^\top \cdot \left[\mathbf{h}'_v \| \mathbf{h}_{P(v,u)}\right]\right) \\ \alpha_{vu}^P &= \text{SOFTMAX}_{u \in \mathcal{N}_v^P}\left(e_{vu}^P\right), \\ \mathbf{h}_v^P &= \sigma\left(\sum_{u \in \mathcal{N}_v^P} \alpha_{vu}^P \cdot \mathbf{h}_{P(v,u)}\right). \end{aligned} \tag{11}$$

where $\mathbf{a}_P \in \mathbb{R}^{2d'}$ is the parameterized context vector specific for P, $\|$ denotes the vector concatenation operator. e_{vu}^P indicates the absolute importance of metapath instance $P(v,u)$ and then is normalized using softmax function. $\mathbf{h}_v^P$ denotes the final representations of metapath P, which is acquired by using the weight factor α_{vu}^P and the representations of all instances.

3) INTER-METAPATH AGGREGATION

After applying intra-metapath aggregation layers, we obtain the vector representation of a single metapath. However, we still need to combine every metapath by using inter-metapath aggregation in order to generate the vector representation of the target node. Consider the set of metapath $\{P_1, P_2, \ldots, P_N\}$ which start with node type $A \in \mathcal{A}$, given a node $v \in \mathcal{V}_A$, we have $|\mathcal{V}_A|$ set of representations $\{\mathbf{h}_v^{P_1}, \mathbf{h}_v^{P_2}, \ldots, \mathbf{h}_v^{P_N}\}$ generated by the intra-metapath aggregation, where N stands for the number of metapaths. However, metapaths are not equally important, we need to apply the attention mechanism to the inter-metapath aggregation.

First, we need to obtain the representations of each metapath across all node $v \in \mathcal{V}_A$, we have:

$$\mathbf{s}_{P_i} = \frac{1}{|\mathcal{V}_A|} \sum_{v \in \mathcal{V}_A} \tanh\left(\mathrm{M}_A \cdot \mathbf{h}_v^{P_i} + \mathbf{b}_A\right) \tag{12}$$

where $\mathrm{M}_A \in \mathbb{R}^{d_m \times d'}$ and $\mathbf{b}_A \in \mathbb{R}^{d_m}$ are learnable parameters.

Then the attention mechanism is used to fuse the metapath specific information into the node representations of v, we have:

$$\begin{aligned} e_{P_i} &= \mathbf{q}_A^\top \cdot \mathbf{s}_{P_i}, \\ \beta_{P_i} &= \text{SOFTMAX}_{P_i \in \mathcal{P}_A}(e_{P_i}), \\ \mathbf{h}_v &= \sum_{P \in \mathcal{P}_A} \beta_P \cdot \mathbf{h}_v^P, \end{aligned} \tag{13}$$

where $\mathbf{q}_A \in \mathbb{R}^{d_m}$ is the parameterized attention vector for node type A, β_{P_i} indicates the importance of P_i, $\mathbf{h}_v$ is the final representations of target node v which is the weighted sum of structural and textual information of every metapaths.

At last, using heterogeneous graph representation learning algorithm above we can obtain the vector representations of the new case and any other node v which are $\mathbf{h}_{new}$ and $\mathbf{h}_v$ respectively. We calculate the probability that the new node and v link together as follows:

$$p = \sigma\left(\mathbf{h}_{new}^\top \cdot \mathbf{h}_v\right) \tag{14}$$

where $\sigma(\cdot)$ is the sigmoid function.

C. TRAINING

The training objective of legal article identification consists of two parts. The first one is to minimize the cross-entropy between predicted charge distribution $\hat{\mathbf{y}}_c$ and the ground-truth distribution $\mathbf{y}_c$:

$$\mathcal{L}_{charge} = -\mathbf{y}_c \log(\hat{\mathbf{y}}_c) - (1 - \mathbf{y}_c)\log(1 - \hat{\mathbf{y}}_c). \tag{15}$$

The other one is to minimize the cross-entropy between predicted distribution and the ground-truth of legal article identification:

$$\mathcal{L}_{article} = -\mathbf{y} \log(\hat{\mathbf{y}}) - (1 - \mathbf{y})\log(1 - \hat{\mathbf{y}}), \tag{16}$$

where $\hat{\mathbf{y}}$ is the result of prediction and $\mathbf{y}$ is the true label.

Considering the two objectives, the final loss function of legal article identification $\mathcal{L}_1$ is obtained by adding $\mathcal{L}_{charge}$ and $\mathcal{L}_{article}$ in the form of joint loss function:

$$\mathcal{L}_1 = \mathcal{L}_{charge} + \mathcal{L}_{article}. \tag{17}$$

As for legal case retrieval, we adopt a negative sampling [33] method to optimize the parameters of the model by minimizing the loss function as follow:

$$\mathcal{L}_2 = -\sum_{(u,v) \in \Omega} \log\sigma\left(\mathbf{h}_u^\top \cdot \mathbf{h}_v\right) - \sum_{(u',v') \in \Omega^-} \log\sigma\left(-\mathbf{h}_{u'}^\top \cdot \mathbf{h}_{v'}\right) \tag{18}$$

where $\sigma(\cdot)$ is the sigmoid function, Ω is the set of node pairs where the two nodes in the pairs are connected, Ω^- is the node

pairs sampled from the node pairs in which the two nodes are not connected.

VI. EXPERIMENTAL SETTINGS

In this section, we introduce the datasets, hierarchy of legal statute, metrics and baseline models we use in the experiments.

A. DATASETS AND HIERARCHY OF LEGAL STATUTE

We use two datasets to evaluate our model, LeCaRD [20] and ELAM [21]. And we adopt three acts in the Chinese judiciary to build LeDSGra, which are the Criminal Law of the People's Republic of China, the Criminal Procedure Law of the People's Republic of China and the Interpretation of the Supreme Court on the application of the Criminal Procedure Law.

LeCaRD consists of 107 query cases and 10,700 candidate cases selected from a corpus of over 43,000 Chinese criminal judgements published by the Supreme People's Court of China. For each query in LeCaRD, there are 100 candidate cases and about 10 relevant cases. The case documents in the corpus have 7 (*key*, *value*) pairs which are shown in the Table 1.

TABLE 1. Descriptions of keys in the corpus document.

Key	Description
ajID	Unique case ID
ajjbqk	Basic case information
cpfxgc	Court analysis
pjjg	Judgment
qw	Full text
writID	Unique document ID
writName	Document title

The basic case information is a brief summary of case, and thus the vector representations of it function as the initial attributes of a legal case document. Moreover, the basic case information of a case document contains legal article citations of the legal case, and thus we can extract statute citations from the basic case information to build LeDSGra. To achieve this, we mainly use regular expression based patterns, e.g., the pattern $<$ [section or article number] of the [Act] $>$ is used to extract citations such as 'Section 47 of the Criminal Law of the People's Republic of China'.

ELAM consists of 5000 legal case pairs and 8955 legal cases collected from Faxin.[1] Each legal case pair is associated with a matching tag which is 2, 1, or 0 representing match, partially match or mismatch respectively. In our experiment, legal case pairs with label 2 and 1 are regarded as similar case pair. What's more, the text content of each case in the pair includes the basic case information, trial and the application of the law. Given that there are no query or new legal case in the ELAM, structural information of each legal case can be obtained by using regular expression based patterns.

[1]https://www.faxin.cn

The number of levels of the hierarchy in the three acts we adopt is shown in the Table 2.

TABLE 2. Levels number of the hierarchy in the three acts.

	Act1	Act2	Act3
num of Chapters	2	5	0
num of Topics	10	21	24
num of Sections	37	14	24
num of Article	451	290	547

The Act1, Act2, Act3 in the Table 2 represent the Criminal Law of the PRC, the Criminal Procedure Law of the PRC and the Interpretation of the Supreme Court on the application of the Criminal Procedure Law respectively.

B. METRICS

In order to measure the performance of the model, we utilize the precision metrics including P@$k(k = 5, 10)$ and MAP, and the ranking metric NDCG@$k(k = 10, 20, 30)$ for LeCaRD which focuses on the similar case retrieval. As for ELAM which focuses on legal case matching in case pairs, we use Accuracy, Macro-Precision, Macro-Recall, and Macro-F1 as metrics.

C. BASELINES

To demonstrate the effectiveness of our model, we conducted a comparative analysis against multiple network-based and text-based methods. The list of baselines is presented below:

- **BM25** [34] is a classic ranking algorithm in information retrieval based on the probabilistic retrieval framework.
- **LMIR** [35] is a non-parametric information retrieval method that combines document indexing and document retrieval into a single model.
- **TF-IDF** [36] is a numerical statistic that is intended to reflect the importance of a word in a document from a corpus. It can be used as a weighting factor in information retrieval.
- **BERT** [17] is a pre-trained language model which trains on a large scale of unlabeled corpus. BERT model has made breakthrough achievements in the NLP field and been applied to the legal tasks, we use BERT$_{base}$ model in particular.
- **RoBERTa** [37] is a pre-trained language model based on masking strategy of BERT and removes the task of next-sentence prediction.
- **Lawformer** [25] is a Longformer-based pre-trained language model for Chinese legal long documents understanding.
- **BERT-PLI** [23] use BERT model to obtain the semantic relationships and infers the relevance between two cases by aggregating paragraph-level interactions with RNN and attention model. Finally, similarity score is obtained by using an MLP.
- **BERT-LF** [38] combines legal facts, topic distribution and legal entity to generate better legal document case

TABLE 3. Results(%) of legal case retrieval on ELAM and LeCaRD.

Dataset	ELAM				LeCaRD			
Metrics / Models	Acc.	P.	R.	F1	Acc.	P.	R.	F1
BERT	0.688	0.698	0.669	0.672	0.973	0.86	0.86	0.86
roBRETa	0.775	0.844	0.807	0.825	0.958	0.785	0.785	0.785
BERT-PLI	0.707	0.689	0.684	0.688	0.616	0.609	0.604	0.605
Lawformer	0.699	0.723	0.712	0.709	0.885	0.407	0.407	0.407
Hier-SPCNet	0.646	**0.922**	0.504	0.652	0.957	0.792	0.792	0.792
Our model	**0.867**	0.916	**0.878**	**0.897**	**0.997**	**0.986**	**0.986**	**0.986**

representations. A method of paragraph aggregation based on BERT is used to encode context semantic information and solve the problem of long text.

- **A BERT-based two-stage method** [39] first uses the BM25 ranking function to retrieve top n case candidates and uses the BERT to accurately sort the recalled case candidates.
- **Hier-SPCNet** [19] is a network-based approach which considers the precedent citation network among case documents and the hierarchy of legal statutes.

VII. EXPERIMENT RESULTS AND DISCUSSION

In this section, we first present the performance of legal case retrieval, and then discuss the impact of different components of our model using ablation test. We also conduct a case study to further analyze our model, at last we present the performance of legal law identification and its effect on the legal case retrieval.

A. PERFORMANCES OF LEGAL CASE RETRIEVAL

We conduct experiments on the LeCaRD and ELAM datasets to compare the performance of different baseline models on the task of legal case retrieval. We input the topological relationships and textual contents of the three acts and training cases into the model, which allows us to obtain representations for each node in the LeDSGra. We then utilize the case representations to retrieve similar cases for the cases in the test set.

Since there are predefined query sets in the LeCaRD dataset, we firstly conduct experiments on the common query set in LeCaRD dataset and evaluate the results using traditional retrieval evaluation metrics, including P@5, P@10, MAP and NDCG. The result is shown in the Table 7.

Subsequently, we partition similar case pairs in both LeCaRD and ELAM datasets into training and testing sets. Specifically, the models are trained on 80% of the data and tested on the remaining 20%. The data partitioning method for the LeCaRD dataset follows the approach in the work [20]. For ELAM, we utilize the first 80% of the dataset as the training set and the remaining 20% as the testing set following the original order of ELAM. Accuracy, Macro-Precision, Macro-Recall and Macro-F1 are used to evaluate the experiments. The result is shown in the Table 3.

From Table 7, we can see that the performance of our model significantly exceeds that of other baseline models. This fully demonstrates that combining both text-based and network-based information can significantly improve the performance of the model. Among text-based methods, the BERT model performs the best in terms of P@5, P@10, and MAP metrics. However, it falls short in terms of NDCG@20 and NDCG@30 compared to models specifically for the legal domain, such as BERT-PLI, BERT-LF. This is because these models make improvements tailored to the legal domain, enabling them to provide more accurate results when recommending a larger number of legal cases. Moreover, as a network-based method, Hier-SPCnet demonstrates superior performance in terms of P@5 compared to traditional retrieval methods and some BERT-based methods. However, its performance is poorer in terms of NDCG@10. This implies that Hier-SPCNet, leveraging topological relationships, can quickly identify some correct similar cases. However, it struggles to find more correct cases due to the sparsity of the network. Note that although text-based methods show comparable performance to traditional retrieval methods in terms of NDCG@30, they outperform traditional methods in terms of NDCG@10. This suggests that BERT-based methods can maintain better performance when recommending a limited number of similar cases, which is of great concern in practical scenarios. Furthermore, the two-step approach demonstrates better performance across multiple metrics by employing two rounds of filtering. In general, network-based methods are more adept at utilizing the specific structural information to achieve better results when the number of retrieved cases is limited. Therefore, they perform well in terms of the P@5 metric. On the other hand, text-based methods excel at leveraging textual information to extensively find similar cases, hence performing better in terms of NDCG@20 and NDCG@30 metrics. However, our proposed model combines the strengths of both two methods, allowing it to achieve best results on all metrics.

As shown in the Table 3, our model exhibits significantly higher performance than other baseline models on both the ELAM and LeCaRD datasets in most of the metrics. In the ELAM dataset, the Hier-SPCNet model outperforms our model in terms of precision, but it has the lowest accuracy, recall and F1 among all models. This indicates

TABLE 4. Results (%) for ablation study.

Datasets	LeCaRD						ELAM			
Metrics / Model	P@5	P@10	MAP	NDCG@10	NDCG@20	NDCG@30	Acc	P	R	F1
SN	0.585	0.494	0.722	0.731	0.759	0.780	0.698	0.952	0.569	0.712
ON	0.118	0.111	0.152	0.131	0.142	0.179	0.401	0.665	0.178	0.281
OT	0.688	0.571	0.834	0.842	0.866	0.88	0.744	0.875	0.712	0.785
w/o P_1	0.682	0.544	0.760	0.822	0.814	0.812	0.743	0.817	0.784	0.80
w/o P_2	0.351	0.329	0.380	0.40	0.429	0.476	0.462	0.662	0.37	0.475
w/o P_3	0.279	0.289	0.388	0.353	0.443	0.504	0.556	0.70	0.568	0.627
w/o P_4	0.757	0.642	0.886	0.909	0.919	0.921	0.601	0.704	0.677	0.690
Our model	**0.794**	**0.653**	**0.988**	**0.99**	**0.994**	**0.994**	**0.867**	**0.916**	**0.878**	**0.897**

that it performs poorly in balancing precision and recall, which further emphasizes that network-based methods can accurately retrieve a portion of positive instances relying on structural information, but struggle to identify other positive instances which need more textual information. Among all the text-based methods, the Lawformer and roBERTa models achieve better results, which is probably because their structure fits the ELAM dataset. Note that the recall and accuracy are the same on the LeCaRD dataset. This is because we sort all candidate cases based on the calculated probabilities of similarity, and then select the top number of true positive cases as predicted positives. Therefore, for each false positive, there is a corresponding false negative, resulting in equal recall and accuracy rates. In general, the performance of our model on ELAM is worse than that of LeCaRD. This disparity may be attributed to the sparser network structure of ELAM, which has limited structural information.

Our model which combines both network-based and text-based methods gains best performance among strange baselines. This demonstrates the advantages of our model in the task of legal case retrieval.

B. ABLATION TEST

To validate the importance of each components in our model, we conduct a ablation test on different variants of our model. Specifically, we remove some parts of the model separately and validate the effectiveness of the variant. Table 4 displays the performance of different variant models on the ELAM and LeCaRD datasets. Here SN, which means simple network-based information, refers to the proposed model that merely utilizes Act_1(the Criminal Law of the People's Republic of China) to construct LeDSGra; OT, which means only text-based information, refers to the proposed model without the component for processing network-based information; ON, which stands for only network-based information, refers to the proposed model without the component for processing textual information. Besides, w/o P_1, w/o P_2, w/o P_3, w/o P_4 refer to the proposed model without metapath P_1, P_2, P_3, P_4 respectively, which are defined in section III-B.

From the Table 4, by leveraging both textual and structural information, our model performs better than all the variants, which shows the importance of combining both two sorts of information. Among all the variant models, ON performs the poorest on both two datasets, which indicates that the structural information in the LeDSGra is more significant than textual information. Surprisingly, SN which applies Act_1 performs worse than OT. This is because the Act_1 is intended for determining charges in a case, but cases with the same charge can also exhibit significant dissimilarities. Moreover, the difference between the results of ON and OT reveals that and text-based methods tend to outperform network-based methods in practical application scenarios where networks are generally sparse. Moreover, metapath P_2, which is **document-article-article-document**, is deemed the most crucial among all metapaths, as its removal results in the largest decline in model performance on both two datasets, followed by metapath P_3. This probably because similar articles of law and articles under the same section provide the information on the types of crime in the legal cases which could be more important than the exact charges of cases. Surprisingly, the contribution of P_1 to the model falls short of expectations, which is probably because the practice of citing the same articles of law in similar cases is less prevalent compared to citing similar articles of law or articles under the same section and there could be a big difference between legal cases citing the same articles.

C. CASE STUDY

In this part, we select a representative case to give an intuitive illustration of how the LeDSGra based on legal documents and legal statutes hierarchy help to promote the performance of the model.

As shown in the Fig.5, the main crime of the defendant in the query case was to withdraw money from bank card despite knowing it is boodle obtained by fraud. One of the similar legal cases of the query case is shown in the Fig.6 whose fact descriptions are mainly the process of fraud by the defendant in opening a fake lottery website, and there is only one sentence mentioning the transfer of boodle. Due to the large amount of irrelevant content in the two cases, vector representations of them generated by other models are likely to be irrelevant. Therefore, it is difficult to decide that they are similar cases. However, with the LeDSGra, the model can know that both of the cases

The new query case:
2016 年 4 月 25 日，靳某骗取湖北省随州市商户张正宇人民币 64000 元。当日，靳某找到被告人朱某让其帮忙找人刷卡取现，被告人朱某在明知靳某银行卡内为诈骗所得款的情况下，介绍康某帮助勒将骗取的 64000 元取出。

On April 25, 2016, Jin cheated Zhang Zhengyu, a merchant in Suizhou city, Hubei Province, of 64,000 yuan. On the same day, Jin asked the defendant Zhu for help to get the boodle from his bank card. Although the defendant Zhu knew that the money Jin wanted was obtained by fraud, he still introduced Kang to help Jin to get the boodle.

FIGURE 5. An illustration of the new query case.

2013 年 11 月，被告人刘雪龙，肖某等人，通过网络设立虚假的彩票网站，并以 19.4%的高额赔率作为诱饵吸引被害人注册充值，骗取被害人投注款。张某、吴某注册长沙创讯科技有限公司，由吴某作为公司法定代表人。肖某负责该公司运作经营，刘某负责广告部运作，由广告部员工发布高赔率彩票广告，诱使被害人加入代理部员工 QQ 群，再由代理部员工冒充玩家与被害人聊天，骗取被害人信任后，引诱被害人到该公司彩票网站注册并充值。待被害人将钱汇入指定账户后，肖某立即将被害人投注款转出并且占有。…

In November 2013, defendants Liu, Xiao and others set up a fake lottery website, and used the high odds of 19.4% as a bait to attract victims to register and recharge, and defrauded the victims of betting money. Zhang and Wu registered Changsha Chuangxun Technology Company with Wu as the company's legal representative. Xiao and Liu are responsible for the operation of the company and the advertising department respectively. Advertising department publishes high-odds lottery advertisements to Induce victims to join the agency staff QQ group where agency staff pretended to be players to chat with the victims and lure them to register and recharge. After the victim remitted the money to the designated account, Xiao immediately transferred the victim's betting money and took possession of it....

FIGURE 6. An illustration of the similar case.

cite the Article 312 which is shown in the Fig.7 which is mainly about how to punish offender concealing criminal proceeds. While training, the specific content of Article 312 and legal cases of the same crime fuse to generate the representations fo similar case in the Fig.6 through metapath **document-article-document**, and thus the representations of the similar case not only include the textual content of itself, but only helpful information to measure similarity with the query case. The representations of the new query case are generated in the same manner when testing, therefore the representations of the two cases are closer with the LeDSGra which indicates that constructing the Legal Documents and Legal Statutes Hierarchy as a heterogeneous graph can improve the performance of the model.

D. PERFORMANCES OF LEGAL LAW IDENTIFICATION

Similarly to real-world scenarios, the query cases in the LeCaRD dataset does not include explicit references to relevant legal articles, instead focusing solely on case descriptions. Therefore, we conduct experiments on LeCaRD dataset to present the performance of legal law identification.

TABLE 5. Performance of legal law identification(Precision and Recall).

k	P.	R.
1	0.598	0.450
2	0.435	0.606
3	0.364	0.736
4	0.299	0.802
5	0.256	0.849
6	0.229	0.909
7	0.203	0.951
8	0.182	0.967
9	0.165	0.977
10	0.149	0.977

Specifically, all query cases in the LeCaRD dataset are utilized as the testing set, while other case documents serve as the training set. In the absence of ground truth labels for the query cases in LeCaRD, we employ an approach combining retrieval of corresponding case judgments from the Wenshu[2] and manual annotation to establish the true

[2]https://wenshu.court.gov.cn

【掩饰、隐瞒犯罪所得、犯罪所得收益罪】明知是犯罪所得及其产生的收益而予以窝藏、转移、收购、代为销售或者以其他方法掩饰、隐瞒的，处三年以下有期徒刑、拘役或者管制，并处或者单处罚金。

Article 312 of Act 1

[Crime of Covering Up or Concealing Criminal Proceeds] Whoever clearly knows that they are criminal proceeds but harbors, transfers, purchases, or sells them, or otherwise conceals them, is to be sentenced to up to three years imprisonment, short-term detention or controlled release, and/or a fine.

FIGURE 7. An illustration of the relevant article of law.

TABLE 6. Performance of legal case retrieval with different value of k.

k	P	R	P@5	P@10	MAP	NDCG@10	NDCG@20	NDCG@30
0	0	0	0.518	0.45	0.691	0.69	0.746	0.788
1	0.598	0.45	0.561	0.474	0.785	0.757	0.794	0.823
2	0.435	0.606	0.507	0.469	0.587	0.618	0.654	0.688
3	0.364	0.736	0.609	0.532	0.688	0.737	0.753	0.766
5	0.256	0.849	0.679	0.590	0.858	0.878	0.898	0.912
9	0.165	**0.977**	**0.794**	**0.653**	**0.988**	**0.99**	**0.994**	**0.994**
10	0.149	**0.977**	0.766	0.650	0.925	0.685	0.722	0.821

TABLE 7. Results(%) of legal case retrieval on the query set of LeCaRD.

Models \ Metrics	P@5	P@10	MAP	NDCG@10	NDCG@20	NDCG@30
BM25	0.406	0.381	0.484	0.731	0.797	0.888
TF-IDF	0.304	0.261	0.457	0.795	0.832	0.848
LMIR	0.434	0.406	0.495	0.769	0.818	0.9
BERT	0.688	0.571	0.834	0.842	0.866	0.88
roBERTa	0.652	0.556	0.777	0.803	0.830	0.846
Lawformer	0.372	0.365	0.459	0.403	0.424	0.459
BERT-PLI	0.32	0.355	0.436	0.743	0.807	0.891
BERT-LF	0.49	0.445	0.592	0.816	0.864	0.919
Two-step method	0.61	0.565	0.662	0.904	0.922	0.958
Hier-SPCNet	0.465	0.366	0.513	0.577	0.585	0.642
Our Model	**0.794**	**0.653**	**0.988**	**0.99**	**0.994**	**0.994**

labels. Subsequently, we evaluate the performance of our model based on these verified ground truth labels.

With different number of legal law identified, the performance of question answering model is illustrated in Table 5, where k represents the number of laws retrieved from the candidate pool.

It can be observed that as the value of k increases, the precision shows a decreasing trend. In contrast, the recall initially increases and eventually stabilizes at 0.977 when $k \geq 9$. This stabilization indicates that a significant portion of legal laws relevant to the query case can be successfully retrieved. The decrease in precision can be attributed to the model retrieving more irrelevant laws than relevant ones for the query cases. Also, the number of relevant articles varies greatly in each case, so this we argues that recall is more important in this task.

In this article, we posit that obtaining structural features of query cases in LeDSGra through legal law identification is crucial for generating appropriate vector representations of query cases. Thus, the performance of the legal law identification directly impacts the overall performance of legal case retrieval. The parameter k plays an important role as it determines the quality of extracted structural features. To elucidate the influence of legal law identification on the performance legal case retrieval, we conduct legal case retrieval with different values of k. The result is shown in the Table 6.

Based on the table, it is evident that the model achieves its best performance in legal case retrieval when $k = 9$, corresponding to the point where recall of legal law identification first reaches its maximum. When $k \leq 9$, the performance of legal case retrieval generally improves as the recall rate increases, except for a slight decline in performance when $k = 2$. However, as $k \geq 9$, the model's performance starts to decline. This suggests that the optimal performance of the model aligns with the first point

of maximum recall in legal law identification. This outcome can be attributed to the heterogeneous graph architecture of LeDSGra, which effectively assigns more weight to accurate information. Consequently, as the recall rate increases, more correct structural information is incorporated, leading to overall improved performance. Nevertheless, when the recall rate reaches its maximum, it becomes crucial to minimize the inclusion of errors. When applying our method on other datasets, setting k to be bigger than the average number of relevant articles is suggested.

VIII. CONCLUSION

In this paper, we propose a novel approach that simultaneously utilizes network-based and text-based information to address three limitations in the task of legal case retrieval: (1) Typical text-based merely consider legal case documents and overlook the significant role of network-based information; (2) Typical network-based methods ignore the rich text-based information in the legal documents and are not meaningfully applicable when legal citation networks are sparse; (3) Existing methods are not applicable in inductive situations. Specifically, our model employs the following three steps: (1) metapath construction; (2) legal article identification; (3) legal case retrieval with a graph representation learning method.

Our model achieves state-of-the-art results on two real-world datasets, ELAM and LeCaRD. And the ablation experiments demonstrate the positive impact of the components in boosting the performance. In the future, we plan to further improve the generalization ability of our model by developing faster methods for generating appropriate vector representations of new cases. We also plan to conduct further research on the metapaths that have positive effects on the task and incorporate them into our model.

ACKNOWLEDGMENT

Any opinions or conclusions in this article are those of the authors only, and should not be interpreted as an official policy or attitude of the nation or government.

REFERENCES

[1] S. Umamaheswari, J. Kanimozhi, and R. Suhashini, "Building accurate legal case outcome prediction models," in *Proc. 2nd Int. Conf. Advancements Electr., Electron., Commun., Comput. Autom. (ICAECA)*, Jun. 2023, pp. 1–6.

[2] D. Hsieh, L. Chen, and T. Sun, "Legal judgment prediction based on machine learning: Predicting the discretionary damages of mental suffering in fatal car accident cases," *Appl. Sci.*, vol. 11, no. 21, p. 10361, Nov. 2021.

[3] S. Gao, Y. Li, F. Ge, M. Lin, H. Yu, S. Wang, and Z. Miao, "Match and retrieval: Legal similar case retrieval via graph matching network," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2023, pp. 227–234.

[4] P. Bhattacharya, K. Ghosh, A. Pal, and S. Ghosh, "Legal case document similarity: You need both network and text," *Inf. Process. Manage.*, vol. 59, no. 6, Nov. 2022, Art. no. 103069.

[5] S. Paul, P. Goyal, and S. Ghosh, "Lesicin: A heterogeneous graph-based approach for automatic legal statute identification from Indian legal documents," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, 2022, pp. 11139–11146.

[6] C. Li, J. Ge, K. Cheng, B. Luo, and V. Chang, "Statute recommendation: Re-ranking statutes by modeling case-statute relation with interpretable hand-crafted features," *Inf. Sci.*, vol. 607, pp. 1023–1040, Aug. 2022.

[7] I. C. Sabo, T. R. Dal Pont, P. E. V. Wilton, A. J. Rover, and J. F. Hübner, "Clustering of Brazilian legal judgments about failures in air transport service: An evaluation of different approaches," *Artif. Intell. Law*, vol. 30, no. 1, pp. 21–57, Mar. 2022.

[8] G. De Martino, G. Pio, and M. Ceci, "Multi-view overlapping clustering for the identification of the subject matter of legal judgments," *Inf. Sci.*, vol. 638, Aug. 2023, Art. no. 118956.

[9] Z. Wang, "Legal element-oriented modeling with multi-view contrastive learning for legal case retrieval," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2022, pp. 1–10.

[10] S. Bi, Z. Ali, M. Wang, T. Wu, and G. Qi, "Learning heterogeneous graph embedding for Chinese legal document similarity," *Knowl.-Based Syst.*, vol. 250, Aug. 2022, Art. no. 109046.

[11] T. Bench-Capon et al., "A history of AI and law in 50 papers: 25 years of the international conference on AI and law," in *Artificial Intelligence and Law*. Cham, Switzerland: Springer, 2012.

[12] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string metrics for matching names and records," in *Proc. KDD Workshop Data Cleaning Object Consolidation*, vol. 3, 2003, pp. 73–78.

[13] S. Kumar, P. K. Reddy, V. B. Reddy, and A. Singh, "Similarity analysis of legal judgments," in *Proc. 4th Annu. ACM Bengaluru Conf.*, Mar. 2011, pp. 1–4.

[14] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2013, pp. 2333–2338.

[15] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, Nov. 2014, pp. 101–110.

[16] Z. Hong, Q. Zhou, R. Zhang, W. Li, and T. Mo, "Legal feature enhanced semantic matching network for similar case matching," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[18] A. Minocha, N. Singh, and A. Srivastava, "Finding relevant Indian judgments using dispersion of citation network," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1085–1088.

[19] P. Bhattacharya, K. Ghosh, A. Pal, and S. Ghosh, "Hier-SPCNet: A legal statute hierarchy-based heterogeneous network for computing legal case document similarity," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1657–1660.

[20] Y. Ma, Y. Shao, Y. Wu, Y. Liu, R. Zhang, M. Zhang, and S. Ma, "LeCaRD: A legal case retrieval dataset for Chinese law system," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2021, pp. 2342–2348.

[21] W. Yu, Z. Sun, J. Xu, Z. Dong, X. Chen, H. Xu, and J.-R. Wen, "Explainable legal case matching via inverse optimal transport-based rationale extraction," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 657–668.

[22] A. Mandal, R. Chaki, S. Saha, K. Ghosh, A. Pal, and S. Ghosh, "Measuring similarity among legal court case documents," in *Proc. 10th Annu. ACM India Compute Conf.*, Nov. 2017, pp. 1–9.

[23] Y. Shao, J. Mao, Y. Liu, W. Ma, K. Satoh, M. Zhang, and S. Ma, "BERT-PLI: Modeling paragraph-level interactions for legal case retrieval," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 3501–3507.

[24] P. Bhattacharya, K. Ghosh, A. Pal, and S. Ghosh, "Methods for computing legal document similarity: A comparative study," 2020, *arXiv:2004.12307*.

[25] C. Xiao, X. Hu, Z. Liu, C. Tu, and M. Sun, "Lawformer: A pre-trained language model for Chinese legal long documents," *AI Open*, vol. 2, pp. 79–84, Jul. 2021.

[26] R.-L. Liu, "A new bibliographic coupling measure with descriptive capability," *Scientometrics*, vol. 110, no. 2, pp. 915–935, Feb. 2017.

[27] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[28] R.-L. Liu and C.-K. Hsu, "Improving bibliographic coupling with category-based cocitation," *Appl. Sci.*, vol. 9, no. 23, p. 5176, Nov. 2019.

[29] Y. Tang, R. Qiu, Y. Liu, X. Li, and Z. Huang, "CaseGNN: Graph neural networks for legal case retrieval with text-attributed graphs," in *Proc. Eur. Conf. Inf. Retr.*, 2024, pp. 80–95.

[30] X. Fu, J. Zhang, Z. Meng, and I. King, "MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proc. Web Conf.*, Apr. 2020, pp. 2331–2341.
[31] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," 2019, *arXiv:1902.10197*.
[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[33] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1–9.
[34] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, *Okapi At TREC-3*. Gaithersburg, MD, USA: NIST Special Publication, 1995.
[35] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, ACM New York, NY, USA, Aug. 1998, pp. 202–208.
[36] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, Jan. 1988.
[37] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[38] W. Hu, S. Zhao, Q. Zhao, H. Sun, X. Hu, R. Guo, Y. Li, Y. Cui, and L. Ma, "BERT_LF: A similar case retrieval method based on legal facts," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–9, Apr. 2022.
[39] J. Zhu, X. Luo, and J. Wu, "A bert-based two-stage ranking method for legal case retrieval," in *Proc. 15th Int. Conf.*, 2022, pp. 534–546.

MENGZHE HEI received the bachelor's degree in data engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2022, where he is currently pursuing the master's degree in management science and engineering. His research interests include natural language processing, data mining, and heterogeneous graph.

QINGBAO LIU received the Ph.D. degree in information system engineering from the National University of Defense Technology, Changsha, China, in 2006. He is currently a Professor with the Science and Technology on Information System Engineering Laboratory, National University of Defense Technology. His current research interests include data mining and machine learning subfields, with a focus on streaming and data engineering.

SHENG ZHANG received the M.S. and Ph.D. degrees in management science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2015 and 2023, respectively. He is currently a Lecturer with the Science and Technology on Information Systems Engineering Laboratory, NUDT. His research interests include natural language processing, deep learning, and data mining.

HONGLIN SHI received the bachelor's degree in data engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2022, where he is currently pursuing the master's degree in management science and engineering. His research interests include natural language processing and data mining.

JIASHUN DUAN received the bachelor's degree in data engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2022, where he is currently pursuing the master's degree in management science and engineering. His research interests include natural language processing and data mining.

XIN ZHANG received the B.S. and Ph.D. degrees in system engineering from the National University of Defense Technology (NUDT), China, in 2000 and 2006, respectively. He is currently a Professor with the Science and Technology on Information Systems Engineering Laboratory, College of Systems Engineering, NUDT. His research interests include cross-model data mining, information extraction, and event analysis.