

Highlights

Leverage Knowledge Graph and Large Language Model for Law Article Recommendation: A Case Study of Chinese Criminal Law

Yongming Chen, Miner Chen, Ye Zhu, Juan Pei, Siyu Chen, Yu Zhou, Yi Wang, Yifan Zhou, Hao Li, Songan Zhang

- Proposed a novel Case-Enhanced Law Article Knowledge Graph (CLAKG) that integrates law articles and historical case data, enhancing the accuracy of law article recommendations.
- Introduced an automated CLAKG construction method that uses a Large Language Model (LLM) to reduce manual input and improve scalability. Developed a closed-loop recommendation system, leveraging LLMs and CLAKG for more accurate law article recommendations while mitigating common issues like hallucinations in LLM outputs.
- Achieved significant improvement in accuracy: The proposed method improved law article recommendation accuracy from 0.549 to 0.694, outperforming baseline models such as BERT, DPCNN, TFIDF-RAG, Graph-RAG and Light-RAG.

Leverage Knowledge Graph and Large Language Model for Law Article Recommendation: A Case Study of Chinese Criminal Law

Yongming Chen^a, Miner Chen^b, Ye Zhu^b, Juan Pei^b, Siyu Chen^b, Yu Zhou^b, Yi Wang^b, Yifan Zhou^b, Hao Li^b, Songan Zhang^a

^aGlobal Institute of Future Technology, Shanghai Jiao Tong University, Shanghai, 200240, Shanghai, China

^bDonghua University, Shanghai, 201620, Shanghai, China

Abstract

Court efficiency is vital for social stability. However, in most countries around the world, the grassroots courts face case backlogs, with decisions relying heavily on judicial personnel's cognitive labor, lacking intelligent tools to improve efficiency. To address this issue, we propose an efficient law article recommendation approach utilizing a Knowledge Graph (KG) and a Large Language Model (LLM). Firstly, we propose a Case-Enhanced Law Article Knowledge Graph (CLAKG) as a database to store current law statutes, historical case information, and correspondence between law articles and historical cases. Additionally, we introduce an automated CLAKG construction method based on LLM. On this basis, we propose a closed-loop law article recommendation method. Finally, through a series of experiments using judgment documents from the website "China Judgements Online", we have improved the accuracy of law article recommendation in cases from 0.549 to 0.694, demonstrating that our proposed method significantly outperforms baseline approaches.

Keywords:

law article recommendation, Large Language Model, Knowledge Graph

1. Introduction

In a modern rule-of-law society, the efficiency of court judgments is crucial for ensuring economic growth and social stability[1–3]. Meanwhile, Xvguang [4] revealed that the current backlog of cases in courts is enormous, which means the workload

of judicial personnel is heavy. Wu [5] showed that there is a great need for intelligent tools to assist judicial personnel in improving the efficiency of case adjudication, since intelligent tools boost judicial efficiency by categorizing similar cases, speeding up research, and staying updated with new laws, ensuring faster and more consistent decisions.

Law article recommendation entails predicting the relevant law articles applicable to a case based on its factual description. With the advancement of computer science, many data science techniques have been applied to this task, greatly improving efficiency. Classic law article recommendation techniques, such as LSTM[6], TextCNN[7], GRU[8], RCNN[9], HAN[10], and DPCNN[11], are based on traditional machine learning methods. These methods focus solely on the correspondence between case facts and law article ids, while overlooking the semantic information in law articles. Moreover, these methods are highly susceptible to the effects of insufficient data and imbalanced data labels. Therefore, there is a need to solve above problems.

Some researchers utilized Large Language Models (LLMs) to complete the task of law article recommendation[12–14]. By using a large LLM, users can not only retrieve the law article in numeric and textual form, but also investigate the rationale behind its applicability to the case by asking follow-up questions. Moreover, users have the capability to probe into the rationale for the non-selection of other analogous provisions. However, Dahl, Magesh, Suzgun, et al. [15] found that directly using LLMs is prone to hallucinations, generating incorrect law article information. Therefore, it is necessary to introduce a fully accurate legal information and case knowledge base for auxiliary analysis.

To mitigate the issue of hallucination, researchers introduced Retrieval-Augmented Generation (RAG) technology to enhance the generation capability of LLMs, by retrieving relevant information from external knowledge bases[16, 17]. Currently, RAG technology has been applied in fields such as code generation[18], autonomous driving[19], and enterprise management[20]. However, research on RAG methods in the domain of law article recommendation remains relatively underexplored. RAG is divided into two parts: retrieval and generation. In academic discussions, greater attention is often given to the retrieval component of RAG, whereas the generation pro-

cess is predominantly handled by LLMs. Classic retrieval methods include TFIDF-based retrieval[21], BM25-based retrieval[22], and retrieval methods based on frozen BERT[23]. However, the above methods mainly focus on the word level and rarely use macro semantics for matching.

To deal with that problem, Chaudhri, Baru, Chittar, et al. [24] proposed that the Knowledge Graph (KG), an unstructured database that uses a directed graph structure to store data, is highly suitable to serve as a database for RAG technology. Pan et al. [25] have pointed out that the integration of KGs with LLMs shows great potential for future research. Edge et al. [26] have utilized LLMs to automatically construct knowledge graphs, which are then used to enhance the generation capabilities of LLMs. This method performs well in terms of comprehensiveness and diversity of answers. However, the KGs generated using this method often have complex structures and lack a fixed schema, making them difficult for users to modify, such as adding new cases or removing outdated ones.

To address the above issues, we introduce knowledge graph technology and large language model technology to improve the output accuracy in law article recommendation tasks. Initially, we pre-design the schema of a Case-Enhanced Law Article Knowledge Graph (CLAKG) capable of simultaneously storing current law articles and historical cases information as a database. On this basis, we also introduce an automated CLAKG construction method based on LLM. Furthermore, we design a closed-loop human-machine collaboration method to complete the law article recommendation task. To demonstrate the effectiveness of our proposed method, we constructed a dataset of Chinese criminal law judgments. We compared the method introduced in this paper with approaches based on BERT[23], GRU[8], DPCNN[11], and several RAG approaches such as TFIDF-RAG[21], Graph-RAG[27] and Light-RAG[28]. Finally, we conducted an ablation study of our method.

The remainder of this paper is organized as follows. Section 2 reviews related works about law article recommendation, RAG technology and knowledge graphs in legal domain. Section 3 proposes a closed-loop law article recommendation framework. Section 4 utilizes the method from the third section to construct the CLAKG and compares the accuracy of this method with other baseline approaches. Lastly, the conclusions and

future work are highlighted in Section 5.

2. Related Work

2.1. Law Article Recommendation

Early approaches to law article recommendation treated the task as a text classification or retrieval problem. Classical methods relied on keyword matching or manual feature engineering, but these struggled with the complex language of statutes. With the rise of deep learning, researchers applied neural models like LSTMs[6] and CNNs[7] to match case fact descriptions with relevant law articles. These models improved over bag-of-words baselines by capturing sequential and local context features. Transformer-based language models further advanced the field of BERT[29] and its legal-domain adaptations (e.g., LegalBERT[30]) enabled richer semantic representations of legal text. For example, Chalkidis et al.[31] showed that a BERT-based classifier can predict applicable European human rights articles more accurately than traditional models, while Aletras et al.[32] demonstrated early on the feasibility of data-driven prediction of court decisions and their pertinent articles. Despite these gains, most prior models formulate law article recommendation as mapping facts to article IDs, without directly leveraging the content of the law articles themselves. This focus solely on fact-to-ID correspondence makes them heavily data-dependent and sensitive to label imbalance in training sets[33]. In practice, it also limits interpretability – the models do not explain *why* a recommended article is relevant, since the law articles are not explicitly used in the reasoning process. These limitations have motivated explorations into techniques that can incorporate external legal knowledge and provide better justification for recommendations.

2.2. LLMs and Retrieval-Augmented Generation Techniques

The advent of large language models (LLMs), such as GPT-style transformers, has opened new avenues for law article recommendation. Rather than training a specific classifier, one can prompt an LLM with a case’s facts and ask for relevant law articles. Recent works have started to investigate this approach[12–14], noting that LLMs

can, in principle, leverage their vast pretrained knowledge to identify relevant law articles and even generate explanations. In particular, Shui et al.[14] evaluate various state-of-the-art LLMs on legal judgment prediction subtasks (including relevant law article prediction), finding that while LLMs exhibit promising zero-shot capabilities, they still lag behind task-specific models on accuracy. A major concern with directly using LLMs for legal guidance is **hallucination**: the model may produce plausible-sounding but incorrect or nonexistent legal references. Dahl et al.[15] provide a sobering analysis of this issue, showing that even advanced LLMs frequently fabricate legal citations or statutes, which is unacceptable in high-stakes domains like law. To mitigate such hallucinations, researchers are turning to Retrieval-Augmented Generation (RAG) techniques. RAG combines LLMs with information retrieval: before or during text generation, the system retrieves relevant documents (e.g. law articles or case precedents) and conditions the LLM on that retrieved evidence. By grounding responses in actual law articles, RAG can significantly improve factual accuracy and trustworthiness of the model’s output. This approach was originally popularized in open-domain QA systems[16] and has since been applied to legal NLP as well. For instance, a RAG-based legal assistant might fetch the text of candidate law articles given a case description, and the LLM then quotes or summarizes those law articles when recommending applicable law, thereby rooting its answer in verifiable sources. Key components of RAG include a retriever and a generator. In the legal domain, classic IR methods like TF-IDF[21] or BM25[22] have been used to retrieve statute candidates by keyword overlap, while more recent approaches use neural retrievers (e.g. dense embeddings). Borgeaud et al.[23] showed that even a frozen BERT-based retriever can substantially improve generation quality by supplying relevant text from a large corpus. However, naive retrieval based on surface text may miss conceptual matches. Legal phrasing varies, and a model might need to retrieve statutes using synonymous terms or implied connections, something neural retrieval aims to handle better than lexical methods. Overall, RAG represent a promising direction for law article recommendation, as they combine the reasoning ability of generative models with the grounding of information retrieval, thereby reducing hallucinations and improving the transparency of the recommendation.

2.3. Knowledge Graphs in the Legal Domain

Beyond unstructured text retrieval, Knowledge Graphs (KGs) have emerged as valuable resources for legal information retrieval and recommendation. A legal knowledge graph represents entities such as law articles, cases, legal concepts, and their relationships (e.g., a case *cites* a statute, or a chapter *includes* certain articles) in a structured form. This graph structure can complement text-based methods by capturing the domain’s ontology and the interconnections between law articles. Prior works have constructed legal knowledge graphs to assist various tasks. For example, law-specific knowledge bases have been used to integrate charge definitions and article relationships into prediction models, yielding performance gains and better interpretability in tasks like judgment prediction[34, 35]. More recently, Chaudhri et al.[24] argued that KGs are well-suited as the underlying database for RAG systems: instead of retrieving free text, one can query a knowledge graph to retrieve facts or linked nodes, ensuring that the generation is grounded in a consistent knowledge structure. In the legal domain, a KG could be queried to find statutes related to certain legal concepts or past cases with similar fact patterns. One advantage of KGs is that they store high-level associations beyond lexical similarity – for instance, a graph can link a section of law to broader categories or to precedent cases, enabling the recommendation system to recall relevant law articles even if a case’s description uses different wording. There is growing interest in integrating KGs with LLMs[25]. Pan et al. [25] outline a roadmap for unifying the strengths of symbolic knowledge graphs and distributed language models, noting that KGs can provide factual grounding for LLMs while LLMs can help populate and expand KGs from raw texts. In the context of law, researchers have begun designing hybrid systems that use KGs alongside LLMs. Chen et al.[33], for instance, construct a case-enhanced law article KG that links statutes with relevant past cases, and then use an LLM-based pipeline to recommend articles by traversing the graph for pertinent nodes (law articles) and using the LLM to generate the final recommendations and explanations. Such approaches have demonstrated improved accuracy over text-only baselines, highlighting that the structured knowledge in KGs can effectively guide the retrieval and generation process. Meanwhile, Edge et al.[26] explore automatic construction of knowledge graphs using LLMs (generating graph triples from unstruc-

tured legal text) and using those graphs to enhance answer generation. This suggests a future where legal KGs and LLMs continuously enrich each other: the KG offers a source of truth to reduce LLM errors, and the LLM helps keep the KG up-to-date with new insights. The integration of knowledge graphs in legal AI is still an emerging area, and challenges remain, such as ensuring the completeness and currency of the graph. Nonetheless, the combination of KGs with retrieval-augmented LLMs holds great promise for building more reliable and interpretable law article recommendation systems in the coming years.

3. Method

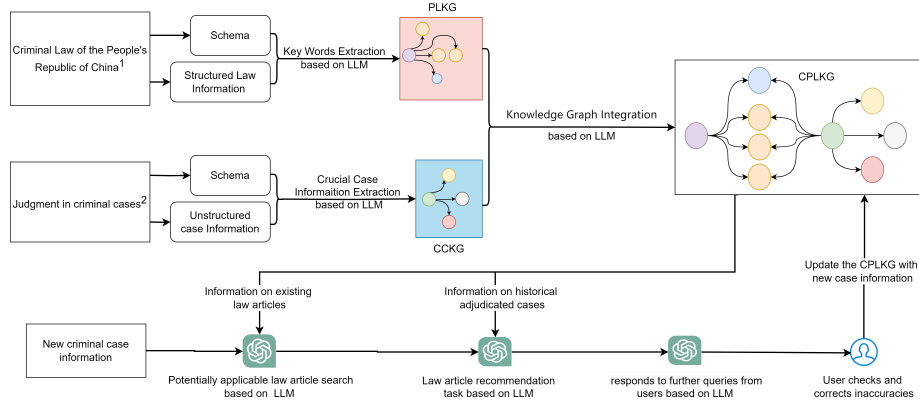


Figure 1: The pipeline of CLAKG. The Case-Enhanced Law Article Knowledge Graph (CLAKG) is composed of the Law Article Knowledge Graph (LAKG) and the Adjudicated Case Knowledge Graph (ACKG).

The pipeline of CLAKG-based law article recommendation is shown in Figure 1. We first establish a knowledge graph based on the designed schema. Initially, nodes and relationships are extracted from law articles and judgments in cases to construct the Law Article Knowledge Graph (LAKG) and the Adjudicated Cases Knowledge Graph (ACKG). Subsequently, by utilizing a Large Language Model (LLM), these two graphs are integrated to form CLAKG. This part is introduced in Section 3.1. Building upon this, we have developed a human-machine Collaborative law article recommendation framework. Users describe new case information, and the LLM would provide potentially applicable law articles based on keyword matching and graph embedding

techniques. Moreover, the system can provide similar historical case information from the CLAKG, followed by providing most relevant law articles that match the new case using LLM. This part is introduced in Section 3.2. Following the above process, the user reviews the law articles provided by the system and corrects any inaccuracies. The new case information is used to update the CLAKG.

3.1. Case-Enhanced Law Article Knowledge Graph Construction

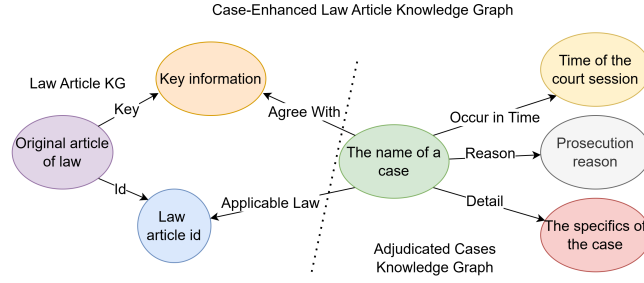


Figure 2: Schema of CLAKG. Ellipses of different colors represent different entity types.

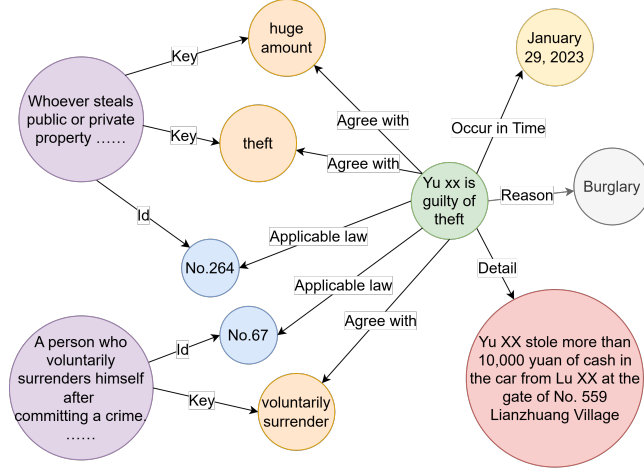


Figure 3: Example of CLAKG.

As shown in Figure 2, the Case-Enhanced Law Article Knowledge Graph (CLAKG) is composed of the Law Article Knowledge Graph (LAKG) and the Adjudicated Case Knowledge Graph (ACKG). The LAKG includes original article of law, law article id, and key information. The key information is extracted from the original article of law

Table 1: Entity types in Case-Enhanced Law Article Knowledge Graph

Entity Type	Description
Original article of law	An original article from the field of law, such as criminal law or traffic law, etc
Key information	Key information in law article extracted by large language model
Law article id	The article number in the law
The name of a case	Case name on the judgment documents
Time of the court session	The time when a court makes a decision in a case
Prosecution reason	Prosecution reason on the judgment documents
The specifics of the case	Case information extracted using a large language model

using a LLM. The ACKG contains the name of a case, time of the court session, prosecution reason, and the specifics of a case. The description of the case process is derived from the content of the judgment documents processed using a LLM.

Based on the reference law articles written in the judgment documents, it is possible to link the name of a case node with law article id nodes. We analyze the correlation between case information and the corresponding key information of applicable law articles using a LLM and ultimately select up to five of the most relevant keywords for connection. To ensure the consistency and accuracy of the knowledge graph, we ignore any keywords that do not exist in the graph output by the large language model. Tables 1 and 2 provide the details of all types of nodes and relationships in the CLAKG.

Table 2: Relationship types in Case-Enhanced Law Article Knowledge Graph

Relation Type	Description
Key	A law article contains several pieces of key information
Id	A law article has a unique law article id corresponding to it
Agree With	A case agree with several pieces of key information
Applicable Law	The law mentioned in the judgment documents that matches the case
Occur in Time	The verdict in a case occurs on a fixed date
Reason	Case Documents contain the reason of lawsuit in the case
Detail	The name of a case corresponds to the specific information of the case

3.2. LLM and CLAKG Based Law Article Recommendation Framework

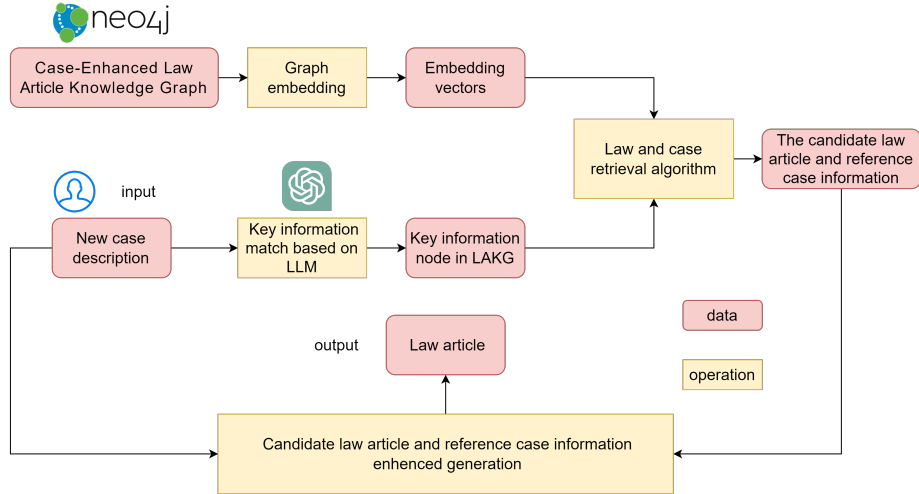


Figure 4: The LLM and CLAKG based law article recommendation framework.

The LLM and CLAKG based law article recommendation framework is shown in Figure 4. Before the start of the law article recommendation task, the CLAKG constructed in Section 3.1 needs to undergo graph embedding preprocessing using the Relational Graph Convolutional Network (RGCN) model mentioned in Section 3.2.1. The process of law article recommendation begins with the user inputting a newly emerged case. In Section 3.2.2, a LLM is used to match the k most relevant keywords to this new case (with $k=8$ as selected in this paper). Based on this, in Section 3.2.3, a candidate

law article retrieval algorithm is employed to retrieve q candidate law articles that may be suitable for the new case (with $q=5$ as chosen in this paper). In Section 3.2.4, the information from the new case input by the user is integrated with the candidate law articles and historical case information obtained in Section 3.2.3 into a prompt, which is then passed to the LLM to complete the law article recommendation task.

3.2.1. Graph Embedding on CLAKG with RGCN

Graph embedding involves utilizing the topological structure information in knowledge graphs to assign embedding vectors to each node within the graph. The Graph Convolutional Network (GCN) model, initially proposed by Kipf and Welling [36], was the first graph neural network used for graph embedding. Schlichtkrull et al. [37] improved upon the GCN, introducing the Relation Graph Convolution Neural Networks (RGCN), which assigns different weights to different relationships, enhancing the quality of graph embeddings.

The node embedding vector update formula for the RGCN model is shown in Equation 1.

$$h_i^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)} \right) \quad (1)$$

where $h_i^{(l)}$ represents the embedding vector of node i in the l -th layer, N_i^r represents all neighboring nodes of node i that have relationship r . $c_{i,r}$ is a hyperparameter, which in this paper is taken as $|N_i^r|$. Both $W_r^{(l)}$ and $W_0^{(l)}$ are parameter matrices.

The graph embedding training task adopted in this paper is link prediction. For the link prediction task, part of the dataset consists of several triples (head node, relation, tail node) selected from the knowledge graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, which are used as positive examples. Another part of the dataset consists of randomly selected entities and relations from the knowledge graph, which are used to construct triples (head node, relation, tail node) that do not actually have a connection in the graph, i.e., fabricated triples, used as negative examples. The training objective of the model is to distinguish whether the given triples truly exist in the graph or are fabricated. The number of positive and negative triples is equal, which helps to prevent bias during model training.

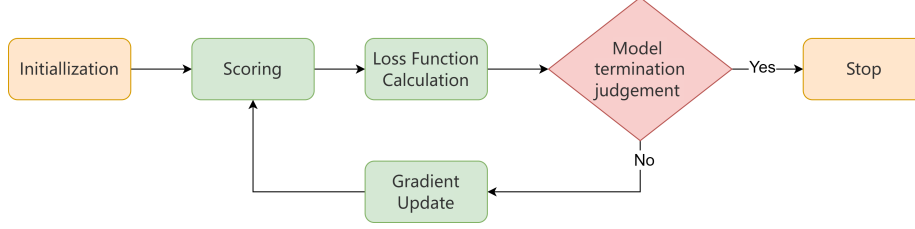


Figure 5: Flowchart of RGCN training process.

The training process of the model is shown in Figure 5. Firstly, the system randomly initializes the embedding vectors for all nodes and relations. Based on this, the system updates the node embeddings using Equation 1. The system then calculates the score based on the DisMult decomposition, as described in Equation 2. This paper selects the cross-entropy function as the loss function, as described in Equation 3. When the triple is a positive example, meaning the information truly exists in the knowledge graph, the value of y is 1. When the triple is a negative example, meaning the information does not exist in the graph, the value of y is 0.

$$f(s, r, o) = e_s^T R_r e_o \quad (2)$$

where e_s and e_o represent the embedding vectors of the nodes s and o , and R_r is the embedding vector of the relation r .

$$\text{Loss} = - \frac{1}{(1 + \omega)|\hat{\mathcal{E}}|} \sum_{(s,r,o,y) \in \mathcal{T}} \left[y \log l(f(s, r, o)) + (1 - y) \log (1 - l(f(s, r, o))) \right] \quad (3)$$

where \mathcal{T} represents all the datasets, and l is the logistic activation function, as described in Equation 4.

$$l(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \quad (4)$$

3.2.2. Matching Key Information Nodes in CLAKG with LLM

In this section, we aim to match the relevant key information nodes within CLAKG based on current case information provided by users. To facilitate this process, we have

designed prompts for the LLM, which include the expert role, task description, details of the newly reported case, known key information nodes, and example outputs. The prompts for LLM in the task of matching key information nodes in the CLAKG related to new case information is shown in Table 3.

Table 3: **Prompts for LLM in the task of matching key information nodes in the CLAKG related to new case information.**

Component	Details
Expert Role	Expert in law article analysis
Task Description	Given the available new case information and the following key information nodes, please output 0 – 8 key information nodes most relevant to the case. If you believe there are fewer than 8 relevant key information nodes, there is no need to force the list to reach 8 nodes.
New Case Information	Description of the new case input by the user
Key Information Nodes	all key information nodes in the CLAKG
Precautions	Please directly output the key information nodes, separated by semicolons (;) without including any additional content (including unnecessary punctuation marks).
Output Example	Public and private property; multiple thefts; sale; committing a crime; causing damage; seriously disrupting public order; multiple thefts; lawfully performing duties

3.2.3. Law Article and Case Retrieval Algorithm

In this section, we propose an algorithm for retrieving historical case information and candidate law articles by leveraging the aforementioned data to identify the law article most relevant to the current case.

The proposed method for retrieving candidate law article involves traversing all ‘key information’ nodes obtained in 3.2.2 and retrieving the ‘Original article of law’ nodes that are connected to the ‘key information’ nodes via ‘key’ relationships. These nodes are then aggregated into a set container, with all ‘Original article of law’ nodes in

Algorithm 1 Candidate Law Article Retrieval Algorithm

Require: key information node, the embedded vector dictionary obtained in Section 3.2.1

Ensure: 5 Original articles of law node

```
1: Initialize: law_article_distances = [] , law_article_set = {}           ▶ Initializes
   a list to store distances and corresponding law articles and a set to store candidate
   law articles
2: for  $e \in$  key information node do
3:   Case_set.add_node( $x \mid x - [\text{key}] \rightarrow e$ )   ▶  $x$  is ‘Original article of penal law’
   node
4: end for
5: for law_article  $\in$  law_article_set do
6:   distance = 0                               ▶ Initializes the distance score
7:   for  $e \in$  key information node do
8:     distance += dis(emb[law_article], emb[e]) ▶ Calculate the distance score
9:     where  $\text{dis}(a, b) = \frac{a \cdot b}{\|a\| \times \|b\|}$  ▶ Cosine similarity formula
10:   end for
11:   law_article_distances.append((distance, law_article))
12: end for
13: sort(law_article_distances)                 ▶ Sort the candidates in descending order
14: return law_article_distances[:5]           ▶ Output the top five candidate law articles
```

the set representing the candidate law article. Given that the number of candidate law articles in the set container may be substantial, it is essential to select those law articles most analogous to the newly reported case for further analysis. To this end, each node in the container is evaluated by calculating the distance scores (cosine similarity) between the ‘Original article of penal law nodes’ embedding vector and the multiple ‘key information’ nodes embedding vector obtained in 3.2.1.

These distance scores are summed to yield a cumulative distance score between each ‘Original article of law’ node and the newly reported case. Ultimately, the top five ‘Original article of law’ nodes with the highest cumulative distance scores are selected

as the candidate law articles, serving as reference points for the LLM in completing the law article recommendation task.

The pseudo-code is shown in Algorithm 1. The time complexity of this algorithm is $O(m \cdot n) + O(m \log m)$, where m is the number of elements in the *law article set* and n is the number of elements in the *key information node set*.

After the above process, by querying the CLAKG for nodes representing ‘the name of a case’ connected to these law articles, we can obtain historical case information for reference.

3.2.4. LLM-Enhanced Law Article Recommendation

Our method intends to incorporate reference case information and the original text of candidate law articles to eliminate the hallucinations of LLMs. After retrieving candidate law articles and relevant historical cases with the method described in Section 3.2.3, The knowledge graph is then queried for key details, and this integrated information helps determine the applicability of the candidate articles. The prompt used for the LLM in this chapter consists of expert roles, task descriptions, information on new criminal cases, reference case information, candidate law articles, and output examples.

After the user describes the case information to the system, the system returns the results generated by the LLM. The user can further inquire about the results provided by the LLM.

4. Experiment

4.1. Datasets

The method proposed in this paper can be applied to any type of law articles in any country. For research convenience, the law articles datasets employed in this section is derived from the *Amendment (XI) to the Criminal Law of the People’s Republic of China*[38], ratified during the 24th session of the Standing Committee of the 13th National People’s Congress. This amendment is systematically structured into two sections: the General Provisions and the Specific Provisions, encompassing a total of

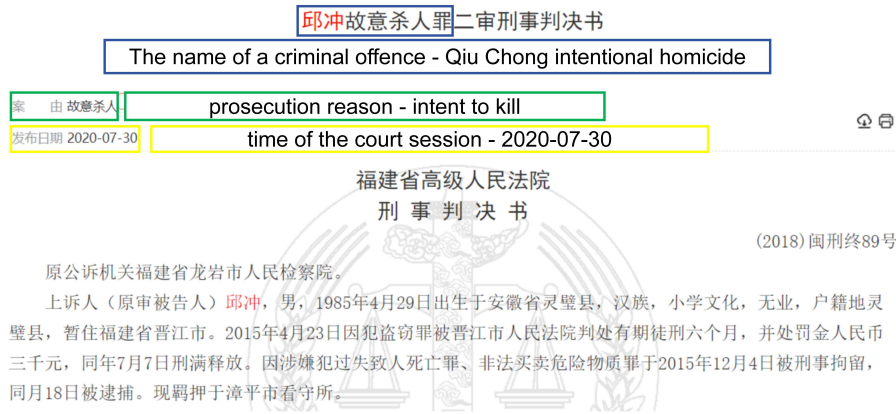


Figure 6: The name of a case, time of the court session, prosecution reason on the China Judgments Online.

452 articles—101 in the General law articles and 351 in the Specific law articles. The CLAKG was constructed utilizing the methodology in Section 3.1.

The criminal case data analyzed in this study were obtained from the China Judgments Online [39] platform. Figure 6 presents one of the cases. In light of the fact that the *Amendment (XI) to the Criminal Law of the People’s Republic of China* came into force on March 1, 2021, all judicial decisions used herein pertain to cases adjudicated after this date. During this period, due to concerns over information security and privacy protection, the number of publicly available criminal case judgments was limited. As a result, certain law articles corresponded with only a limited number of cases, while some provisions had no corresponding cases at all. We curated a dataset comprising 1,170 case records, which collectively address law articles from 85 articles within the Specific Provisions. To optimize the dataset, we divided it in a manner that maximized the diversity of law articles represented in the test set. Specifically, 997 case records were designated for the training set and 173 for the test set. The training set encompassed 49 distinct law articles, whereas the test set covered 85 law articles (*zero shot number* = 36). The Adjudicated Case Knowledge Graph was subsequently constructed based on the training data, following the approach outlined in Section 3.1.

4.2. CLAKG Construction and Graph Embedding

We employed the method described in Section 3.1 to integrate the LAKG and the ACKG, resulting in the CLAKG. The types and numbers of nodes as well as relationships in CLAKG are presented in Table 4.

Table 4: Entity and relation types with their Numbers in CLAKG.

Entity Type	Number	Relation Type	Number
Original article of law	452	Key	7854
Key information	3405	Id	452
Law article id	452	Agree with	997
The name of a case	997	Applicable law	4118
Time of the court session	997	Occur in time	997
Prosecution reason	997	Reason	997
The specifics of the case	997	Detail	997

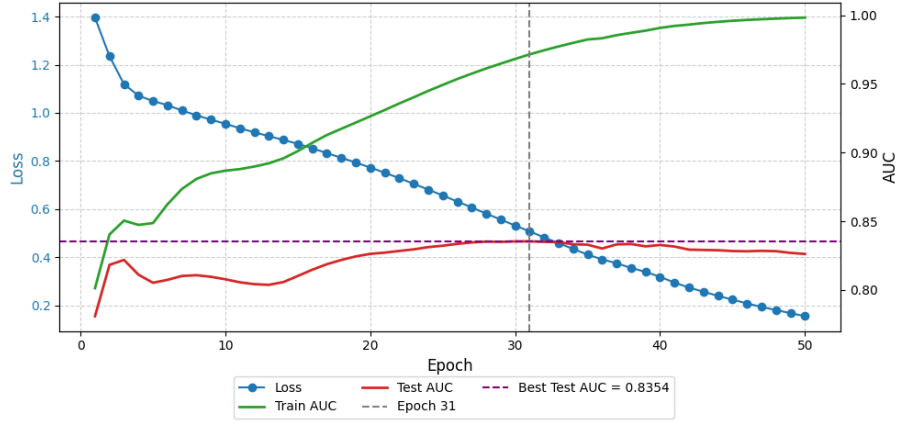


Figure 7: Train loss and AUC over epochs.

We employed the RGCN model described in Section 3.2.1 to perform graph embedding preprocessing on CLAKG. Our hyperparameter choices were as follows: $h_{dim} = 16$, $test_size = 0.2$, $learning_rate = 0.01$, $num_epochs = 50$. We selected the model results corresponding to the highest Test AUC as the final graph embedding outcome.

The Train AUC and Test AUC curves are presented in Figure 7. Upon examining this figure, we observed that the Train AUC exhibited an overall upward trend. The Test AUC increased when $Epoch \leq 31$ and decreased when $Epoch \geq 31$. Consequently, we selected the model results at Epoch 31 as the final graph embedding outcome.

4.3. Case Study

We illustrate the law article recommendation process based on LLM and CLAKG using the case of "Zhang Yue's offenses of bribery and abuse of power" as an example. Initially, the user inputs the details of a new case:

During Zhang Yue's tenure as cultural administrator and grid member of Yongfeng Village, Zhang Yue manipulated residential information to secure illicit benefits, accepting 72,500 yuan in bribes. After the crime was uncovered, he voluntarily admitted his wrongdoing and returned the funds.

The system then retrieves keywords pertinent to the case using the methodology outlined in Section 3.2.2, identifying the following terms: ['accepting bribes', 'abuse of power', 'bribery']. Subsequently, the system locates the corresponding node IDs within the CLAKG and obtains the graph embedding vectors for these nodes as described in Section 4.2. Utilizing the historical case information and the candidate law articles retrieval algorithm specified in Section 3.2.3, the system identifies the candidate law articles and their associated cases.

The system consolidates the above information into a prompt using the method described in Section 3.2.4, and submits it to the LLM to carry out the law article recommendation task. Ultimately, the LLM outputs "Article 385" as the recommended law article. The result produced by the LLM is consistent with the outcome recorded in the court judgment.

4.4. Test Result

4.4.1. Analyzing the Impact of Label Imbalance in Training

We utilized the BERT[23], GRU[8] and DPCNN[11] model for law article recommendation as comparison methods. These methods were compared against the method

proposed in this paper: the LLM(specifically OpenAI’s ChatGPT-4.0) for law article recommendation based on CLAKG (with case information).

Table 5: Model accuracy comparison between classical models and ours

Model	Accuracy
BERT (baseline)	0.289
GRU (baseline)	0.427
DPCNN (baseline)	0.495
LLM + CLAKG (OURS, no case)	0.676

As shown in the Table 5, the accuracy of the BERT (baseline) is only 0.289. Upon analyzing the results provided by the BERT method, we discovered that this method classified all test cases under "Article 133." We believe this could be due to the small size of the training set, an imbalance in training labels[40], or an inappropriate loss function. Even after replacing the cross-entropy loss function with the Focal Loss function, the model still classified all test cases as "Article 133".

The GRU (baseline) model achieved an accuracy of 0.427. GRU, or Gated Recurrent Unit, is a type of recurrent neural network (RNN) that is widely used for sequential data, and this result represents its performance on the task at hand.

The DPCNN (baseline) model, which stands for Deep Pyramid Convolutional Neural Network, performed slightly better, with an accuracy of 0.495. DPCNN is a convolutional neural network architecture designed for text classification tasks, and its accuracy reflects its ability to capture hierarchical patterns in the input data.

The proposed model (LLM+CLAKG) achieved an accuracy of 0.694, which is significantly higher than any one of the classic methods. This indicates that the proposed approach effectively mitigates the impact of the insufficient data and label imbalance in training.

4.4.2. Analyzing the Impact of Hallucinations in LLMs

We utilized the LLM[41] (specifically OpenAI’s ChatGPT-4.0, as well as other LLMs used in different methods), TFIDF-RAG[21], Graph-RAG[27] and Light-RAG[28]

for law article recommendation as comparison method. This method was compared against the method proposed in this paper: the LLM for law article recommendation based on CLAKG (without case information).

Table 6: Comparing proposed method with other RAG method

Model	Accuracy
LLM (baseline)	0.549
TFIDF-RAG (baseline)	0.597
Graph-RAG (baseline)	0.179
Light-RAG(baseline)	0.636
LLM+CLAKG (OURS, no case)	0.676

As shown in the Table 6, the accuracy of the LLM without external legal data is significantly higher than that of the classic model, achieving an accuracy of 0.549. Graph-RAG’s extremely low result may be due to its graph construction or retrieval strategy not aligning well with our dataset, indicating challenges in directly applying that method to this legal domain. The proposed model (LLM+CLAKG) achieved an accuracy of 0.676, which is significantly higher than that of the TFIDF-RAG, Light-RAG, Graph-RAG method. This indicates that the proposed approach effectively mitigates the hallucinations in LLMs. We believe this is because other RAG method only matches at the text level, while the latter matches the new case with keywords at the semantic level. Additionally, the latter utilizes all information within the CLAKG through graph embedding techniques, allowing it to calculate the most relevant law articles from a holistic perspective.

4.4.3. Ablation Study

In this ablation study, we evaluate the impact of incorporating different types of information on the performance of LLMs for law article recommendation tasks. Specifically, we compare the performance of the baseline models with and without additional case and law article information.

We conducted experiments using both the proprietary GPT-4 model and the open-

source Llama-3.1-405b-chat model. The baseline results for GPT-4 showed an accuracy of 0.549, while incorporating case-specific information with CLAKG (Case Law and Statutory Knowledge Graph) improved the accuracy to 0.694. Similarly, for the Llama-3.1-405b-chat model, the baseline performance was 0.619, and adding reference case and statutory information boosted the accuracy to 0.786.

These results clearly demonstrate that the inclusion of relevant information, such as reference cases, significantly enhances model performance across both proprietary and open-source models. This validates the robustness and versatility of our approach, showing its applicability not only to high-performing proprietary models but also to widely available open-source models. The detailed results are summarized in Table 7.

Table 7: Results of different models in the ablation study.

Model	Accuracy
LLM (GPT-4, baseline)	0.549
LLM + CLAKG (OURS, with case)	0.694
LLM (llama-3.1-405b-chat, baseline)	0.619
LLM + CLAKG (llama-3.1-405b-chat, with case)	0.786

This ablation study highlights the significant impact that additional legal context, such as case and law article information, can have on the performance of LLMs for law article recommendation tasks, confirming the effectiveness of our proposed methodology.

5. Conclusion

This paper proposes an efficient law article recommendation approach using a Case-Enhanced Law Article Knowledge Graph (CLAKG) combined with a Large Language Model (LLM). CLAKG integrates law articles and case information. It is characterized by a rich topological structure formed through usage relationships and shared key information found in judgments, which enhances the effectiveness of law article recommendation tasks. The proposed approach integrates LLMs with CLAKG, enabling more accurate recommendations of law articles and related cases by utilizing

macro semantics to mitigate LLM hallucinations. The effectiveness of the proposed approach is verified through comprehensive comparisons with several baseline models on the law article recommendation task. This approach effectively addresses challenges such as insufficient data, imbalanced labels, and LLM hallucinations, leading to a significant improvement in the accuracy of the law article recommendation task.

In future work, we plan to expand the data volume of CLAKG, particularly by incorporating more case data to further enhance the performance of law article recommendation. Additionally, we will explore further applications of the CLAKG-LLM integration to improve the efficiency of court judgments.

References

- [1] Maria Dakolias. “Court performance around the world: a comparative perspective”. In: *Yale Human Rights and Development Law Journal* (2014).
- [2] A. Rizos and P. Kapopoulos. “Judicial Efficiency and Economic Growth: Evidence based on EU data”. In: *Journal* (2021).
- [3] Héctor Fix-Fierro. *Courts, justice, and efficiency: A socio-legal study of economic rationality in adjudication*. Hart Publishing, 2003.
- [4] H. Xvguang. *The Supreme People’s Court released major data on judicial work in the first half of 2024*. <https://www.court.gov.cn/zixun/xiangqing/438521.html>. Accessed: 2024-07-19, 2024.
- [5] Zuyun Wu. *The construction of the rule of law requires the empowerment of science and technology*. 2024. URL: https://www.moj.gov.cn/pub/sfbgw/fzgz/fzgzzqt/fzgzkjxxh/202404/t20240430_498159.html.
- [6] Sepp Hochreiter and Jurgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [7] Yoon Kim. “Convolutional neural networks for sentence classification”. In: *Proceedings of EMNLP*. 2014.

- [8] Kyunghyun Cho et al. “Learning phrase representations using rnn encoder-decoder for statistical machine translation”. In: *Computer Science* (2014).
- [9] Siwei Lai et al. “Recurrent convolutional neural networks for text classification”. In: *Proceedings of AAAI*. Vol. 333. 2015, pp. 2267–2273.
- [10] Zichao Yang et al. “Hierarchical attention networks for document classification”. In: *Proceedings of NAACL*. 2016, pp. 1480–1489.
- [11] Rie Johnson and Tong Zhang. “Deep pyramid convolutional neural networks for text categorization”. In: *Proceedings of ACL*. Vol. 1. 2017, pp. 562–570.
- [12] X. Yang, Z. Wang, Q. Wang, et al. “Large language models for automated q&a involving legal documents: a survey on algorithms, frameworks and applications”. In: *International Journal of Web Information Systems* 20.4 (2024), pp. 413–435.
- [13] S. Ma, C. Chen, Q. Chu, et al. “Leveraging Large Language Models for Relevance Judgments in Legal Case Retrieval”. In: *arXiv preprint arXiv:2403.18405* (2024).
- [14] R. Shui, Y. Cao, X. Wang, et al. “A comprehensive evaluation of large language models on legal judgment prediction”. In: *arXiv preprint arXiv:2310.11761* (2023).
- [15] M. Dahl, V. Magesh, M. Suzgun, et al. “Large legal fictions: Profiling legal hallucinations in large language models”. In: *arXiv preprint arXiv:2401.01301* (2024).
- [16] Patrick Lewis et al. “Retrieval-augmented generation for knowledge-intensive NLP tasks”. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 9459–9474.
- [17] Deng Cai et al. “Recent Advances in Retrieval-Augmented Text Generation”. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. New York, NY, USA: Association for Computing Machinery, 2022, pp. 3417–3419. doi: 10.1145/3477495.3532682.

- [18] Qianwen Gou, Yunwei Dong, Yujiao Wu, et al. “RRGcode: Deep hierarchical search-based code generation”. In: *Journal of Systems and Software* 211 (2024). doi: 10.1016/j.jss.2024.111982.
- [19] X. Dai et al. “VistaRAG: Toward Safe and Trustworthy Autonomous Driving Through Retrieval-Augmented Generation”. In: *IEEE Transactions on Intelligent Vehicles* 9.4 (2024), pp. 4579–4582. doi: 10.1109/TIV.2024.3396450.
- [20] D. E. O’Leary. “The Rise and Design of Enterprise Large Language Models”. In: *IEEE Intelligent Systems* 39.1 (2024), pp. 60–63. doi: 10.1109/MIS.2023.3345591.
- [21] A. Aizawa. “An information-theoretic perspective of tf-idf measures”. In: *Information Processing & Management* 39.1 (2003), pp. 45–65.
- [22] Stephen Robertson and Hugo Zaragoza. “The Probabilistic Relevance Framework: BM25 and Beyond”. In: *Found. Trends Inf. Retr.* 3.4 (2009), pp. 333–389. doi: 10.1561/15000000019.
- [23] S. Borgeaud, A. Mensch, J. Hoffmann, et al. “Improving language models by retrieving from trillions of tokens”. In: *International conference on machine learning*. PMLR, 2022, pp. 2206–2240.
- [24] V. Chaudhri, C. Baru, N. Chittar, et al. “Knowledge graphs: introduction, history and, perspectives”. In: *AI Magazine* 43.1 (2022), pp. 17–29.
- [25] S. Pan et al. “Unifying Large Language Models and Knowledge Graphs: A Roadmap”. In: *IEEE Transactions on Knowledge and Data Engineering* 36.7 (2024), pp. 3580–3599. doi: 10.1109/TKDE.2024.3352100.
- [26] D. Edge et al. “From Local to Global: A Graph RAG Approach to Query-Focused Summarization”. In: *ArXiv abs/2404.16130* (2024).
- [27] Haoyu Han et al. *Retrieval-Augmented Generation with Graphs (GraphRAG)*. 2025. arXiv: 2501.00309 [cs.IR]. URL: <https://arxiv.org/abs/2501.00309>.

- [28] Zirui Guo et al. *LightRAG: Simple and Fast Retrieval-Augmented Generation*. 2024. arXiv: 2410.05779 [cs.IR]. URL: <https://arxiv.org/abs/2410.05779>.
- [29] Jacob Devlin, Chang Ming-wei, Kenton Lee, et al. “BERT: pre-training of deep bidirectional transformers for language understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019, pp. 4171–4186.
- [30] Ilias Chalkidis et al. “LEGAL-BERT: The muppets straight out of law school”. In: *Proceedings of the Findings of EMNLP*. 2020, pp. 2898–2904.
- [31] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. “Neural Legal Judgment Prediction in English”. In: *Proceedings of the 57th Annual Meeting of the ACL*. 2019, pp. 4317–4323.
- [32] Nikolaos Aletras et al. “Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective”. In: *PeerJ Computer Science* 2 (2016), e93.
- [33] Yongming Chen et al. “Leverage Knowledge Graph and Large Language Model for Law Article Recommendation: A Case Study of Chinese Criminal Law”. In: *arXiv preprint arXiv:2410.04949* (2024).
- [34] Haoxi Zhong et al. “How does the Punishment fit the Crime? Multi-task Learning using Multi-views of Law Articles for Judgment Prediction”. In: *Proceedings of AAAI*. 2020, pp. 1250–1257.
- [35] Xiangyang Hu et al. “Few-Shot Charge Prediction with Discriminative Legal Attributes”. In: *Proceedings of COLING*. 2018, pp. 487–498.
- [36] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).
- [37] Michael Schlichtkrull et al. “Modeling relational data with graph convolutional networks”. In: *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*. Springer. 2018, pp. 593–607.

- [38] Standing Committee of the National People's Congress. *Amendment XI to the Criminal Law of the People's Republic of China*. 2020. URL: <http://xingfa.org/11.html>.
- [39] Supreme People's Court of the People's Republic of China. *China Judgments Online*. 2023. URL: <https://wenshu.court.gov.cn/>.
- [40] J. Franklin. "The elements of statistical learning: data mining, inference and prediction". In: *The Mathematical Intelligencer* 27 (2005), pp. 83–85. doi: 10.1007/BF02985802.
- [41] OpenAI. *ChatGPT (4.0)*. Accessed: 2025-02-20, 2023. URL: <https://www.openai.com/chatgpt>.