How Legal Knowledge Graph Can Help Predict Charges for Legal Text

Shang Gao[1,2], Rina Sa[3], Yanling Li[1,2](✉), Fengpei Ge[4], Haiqing Yu[1], Sukun Wang[1], and Zhongyi Miao[1]

[1] College of Computer Science and Technology, Inner Mongolia Normal University, Hohhot 011517, People's Republic of China
wdmxsyf_gs@163.com, {cieclyl,ciecyhq,ciecwsk,ciecmzy}@imnu.edu.cn

[2] Key Laboratory of Infinite-dimensional Hamiltonian System and Its Algorithm Application, Ministry of Education, Hohhot 010000, People's Republic of China

[3] Department of Mathematics and Computer Engineering, Ordos Institute of Technology, Ordos 017000, People's Republic of China
sarina216@163.com

[4] Library, Beijing University of Posts and Telecommunications, Beijing 100876, People's Republic of China
gefengpei@bupt.edu.cn

Abstract. The existing methods for predicting Easily Confused Charges (ECC) primarily rely on factual descriptions from legal cases. However, these approaches overlook some key information hidden in these descriptions, resulting in an inability to accurately differentiate between ECC. Legal domain knowledge graphs can showcase personal information and criminal processes in cases, but they primarily focus on entities in cases of insolation while ignoring the logical relationships between these entities. Different relationships often lead to distinct charges. To address these problems, this paper proposes a charge prediction model that integrates a Criminal Behavior Knowledge Graph (CBKG), called Charge Prediction Knowledge Graph (CP-KG). Firstly, we defined a diverse range of legal entities and relationships based on the characteristics of ECC. We conducted fine-grained annotation on key elements and logical relationships in the factual descriptions. Subsequently, we matched the descriptions with the CBKG to extract the key elements, which were then encoded by Text Convolutional Neural Network (TextCNN). Additionally, we extracted case subgraphs containing sequential behaviors from the CBKG based on the factual descriptions and encoded them using a Graph Attention Network (GAT). Finally, we concatenated these representations of key elements, case subgraphs, and factual descriptions, collectively used for predicting the charges of the defendant. To evaluate the CP-KG, we conducted experiments on two charge prediction datasets consisting of real legal cases. The experimental results demonstrate that the CP-KG achieves scores of 99.10% and 90.23% in the Macro-F1 respectively. Compared to the baseline methods, the CP-KG shows significant improvements with 25.79% and 13.82% respectively.

Keywords: Charge prediction · Easily confused charges · Knowledge graph · Graph attention network

S. Gao and R. Sa—Equal Contribution.

B. Luo et al. (Eds.): ICONIP 2023, LNCS 14452, pp. 408–420, 2024.
https://doi.org/10.1007/978-981-99-8076-5_30

1 Introduction

With the application of information technologies such as Artificial Intelligence (AI) and big data in various scenarios, there is a growing demand for judicial services. To meet the evolving needs of legal judgment by the general public and keep up with the development of the times, researchers have attempted to integrate AI into the field of legal to achieve fairness, intelligence, and efficiency in legal judgments. In China, to promote AI applications such as intelligent information retrieval and Natural Language Processing (NLP) in the legal domain, the Supreme People's Court and the Chinese Information Processing Society of China jointly organized the Challenge of AI in Law (CAIL) competition. This competition has been held continuously for five years since 2018 and has become an important platform for academic exchanges in the field of legal AI. In CAIL 2018, three tasks were set: legal article recommendation, charge prediction, and sentence prediction.

This study focuses on charge prediction within the CAIL competition. Charge prediction is a crucial task in the intelligent judiciary, which involves analyzing factual descriptions in legal cases to predict the charges for defendants. The predicted results can serve as references for judicial personnel, helping to correct the subjective biases of judges and reduce negative impacts caused by intuition and other subjective factors. Predicting Easily Confused Charges (ECC) has always been a research hotspot and challenge in charge prediction. ECC often share similar criminal processes, but they differ in individual criminal behaviors and outcomes. Different charges often lead to different legal judgments and punishments. The examples of ECC are illustrated in Fig. 1. Among them, blue represents the same elements in the two legal cases, and red represents different elements. Although the behavior information and sequence of behaviors of the two defendants were the same, the behavior on the right also resulted in death. The charges and punishments in the two legal cases are completely different. Therefore, differentiating these key elements and the order of elements is crucial for predicting ECC accurately. Existing methods mostly apply mature AI to predict ECC, which often fails to meet many specialized needs of legal professionals. For example, when traditional text classification models like Long Short-Term Memory (LSTM) [1] are applied to charge prediction, they merely encode and process factual descriptions of legal cases, lacking the fine-grained recognition and understanding of tools and behavioral elements within these descriptions. They do not fully utilize the information presented in legal cases.

Charge: **Dangerous Driving**	Charge: **Traffic Accident**
Case Facts: The defendant, ***A***, drove a car with license plate number Ji CXXX along the West Expressway, traveling in the opposite direction from north to south at the segment of Cuigezhuang Road. The defendant's vehicle collided with an oncoming car, resulting in a traffic accident and damage to both vehicles. After the accident, ***A*** fled the scene. According to the examination, ***A′ s*** blood alcohol concentration was 140.30 mg/100ml, indicating drunk driving.	**Case Facts:** The defendant, ***B***, drove a car with license plate number Ji AXXX along the 393 road, traveling from east to west. At the intersection near Sunjiazhuang Village, the defendant's vehicle collided with a three-wheeler driven by ***C***, resulting in a traffic accident where ***C*** sustained fatal injuries despite rescue efforts and both vehicles were damaged. After the accident, ***B*** fled the scene. It was determined that ***B*** bears full responsibility for this accident.

Fig. 1. Easily confused charges comparison

Knowledge Graphs (KGs) are graph-structured data that describe entities and their relationships. Due to their intuitive and rich knowledge representation, KGs have been widely applied in NLP tasks. KGs can generally be categorized into two types: general-purpose KGs and domain-specific KGs. The former contains extensive knowledge, mainly consisting of common-sense knowledge and coarse-grained knowledge. The latter focuses on a specific domain, emphasizing deeper and more specialized knowledge. It incorporates expert experience and industry-specific information, covering fine-grained knowledge. Compared to text, KGs provide more explicit and logical representations, playing a supportive and enabling role in intelligent judiciary. In the legal domain, general-purpose KGs provide limited knowledge. Therefore, researchers have begun to construct domain-specific KGs for addressing legal domain tasks. However, existing legal domain KGs often focus on the entities within legal cases and often overlook the logical relationships between legal entities, that different behavioral relationships between entities can lead to different charges and varying punishments. Therefore, for the task of predicting ECC, it is necessary to construct a novel domain-specific KG that highlights key elements and sequential information of legal cases.

To address these challenges, this paper proposes a Charge Prediction method called CP-KG, which integrates a Crime Behavior Knowledge Graph (CBKG). The contributions of this work are summarized as follows:

(1) We constructed a novel domain-specific knowledge graph in the field of legal, known as the CBKG. It encompasses a wealth of legal entities and relationships, enabling a clear depiction of the criminal process associated with ECC.
(2) We extracted Key Elements (KE) and Case Subgraphs (CS) from legal cases, which supplements the model with fine-grained legal knowledge and enhances its ability to comprehend ECC.
(3) Experimental results on two charge prediction datasets consisting of real legal cases demonstrate that the CP-KG achieves Macro-F1 scores of 99.10% and 90.23% respectively. Moreover, it significantly outperforms the baseline methods, achieving approximately 25.79% and 13.82% improvement in the Macro-F1 metric.

2 Related Work

2.1 Charge Prediction

With the development of deep learning, it has achieved successful applications in NLP in recent years. Inspired by this, researchers have attempted to apply deep neural networks to the task of charge prediction, aiming to improve accuracy by extracting deep semantic information from legal cases. Jiang et al. [2] treated charge labels as supervision and used deep reinforcement learning to extract key factual fragments from case facts, enhancing case representations and improving model performance. Zhong et al. [3] proposed a topological multi-task learning framework that models the explicit dependency relationships among three sub-tasks: legal article recommendation, charge prediction, and sentence prediction. Yang et al. [4] introduced the Multi-Perspective Bi-directional Feedback Network (MPBFN) with word collocation attention mechanism to fully utilize the topological dependencies among multiple task results, effectively improving judgment prediction. Zhao et al. [5] employed a reinforcement learning method to extract sentences containing criminal elements in cases, simulating the process of judgment in real-world scenarios.

2.2 Legal Domain Knowledge Graph

The construction of a legal domain KGs based on legal cases is beneficial for various important legal tasks. Chen et al. [6] treated criminal charges and keywords from charge description articles as entity nodes and defined four types of relationships. They constructed a charge knowledge graph by using a relationship classifier to determine the relationships between entities. Chen et al. [7] addressed the issue of dispersed knowledge and inconvenient queries in the judicial domain by constructing a knowledge graph based on a legal case. They preprocessed the legal case using the Language Technology Platform (LTP). Then, they organized and compiled it using the Neo4j graph database to build a case knowledge graph. Chen [8] proposed a knowledge graph construction method focused on criminal behavior. They utilized the LTP and a pre-built legal dictionary to extract elements from legal cases and then extracted triples of criminal behavior entity relationships with the assistance of Chinese grammar rules. Guo [9] introduced a causation graph based on legal cases, using dependency syntactic analysis and regular expression matching to obtain a relationship between events. Chen et al. [10] developed an information extraction model specifically for criminal cases to construct a drug-related case graph. Hong et al. [11] focused on judicial charges of "motor vehicle traffic accident liability disputes" and defined 20 entity types and 9 relationship types. They used deep learning to extract entities and relationships and construct a knowledge graph for traffic cases. To address the interpretability issue in sentencing prediction, Wang et al. [12] focused on drug trafficking cases as their research subject. Under the guidance of domain experts, they designed concepts and relationships within the case factual descriptions and extracted triples from the case facts based on the knowledge graph ontology.

Although these methods have made some progress in the task of charge prediction, there are still challenges that need to be addressed. Firstly, deep learning cannot accurately identify behavior within factual descriptions at a fine-grained level. This limitation results in poor model understanding for ECC. Secondly, existing KGs tailored to the legal domain often focus on the entity within factual descriptions while neglecting the behavioral relationship between legal entities. This limitation hinders the accurate identification of the criminal process associated with ECC, leading to lower prediction performance.

3 Methods

In this section, we detail the construction process of the CBKG and the various components of the CP-KG. Firstly, in Sect. 3.1, we introduce the process of constructing the CBKG. Next, in Sect. 3.2, we discuss the extraction process of KE and CS, while Sect. 3.3 covers the encoding process of KE and CS. Moving forward, Sect. 3.4 presents the encoding process of Fact Descriptions (FD) in legal cases. Finally, in Sect. 3.5, we describe the process of feature fusion and predict charge.

3.1 Overview

The CP-KG model extracts features and fuses them based on the input legal cases. Then, the resulting information is fed into a Fully Connected Layer to predict the defendant's charges. The specific process is illustrated in Fig. 2. Firstly, the FD of the legal cases and the CBKG are used as input data and passed to the Extract Module, which obtains KE from the FD and CS of the CBKG. Subsequently, TextCNN and GAT are employed to encode KE and CS. Finally, the representations of the FD, KE, and CS are fused to predict the charges. Detailed descriptions of these modules will be provided in the following sections.

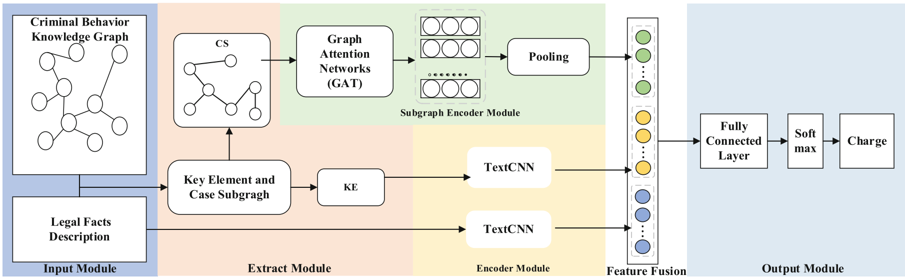

Fig. 2. The structure of the CP-KG model

3.2 Criminal Behavior Knowledge Graph (CBKG)

Collect Legal Cases. The data used for constructing the CBKG is sourced from the criminal legal cases available on China Judgments Online[1]. We have selected eight categories of ECC, namely: intentional injury, intentional homicide, dangerous driving, traffic accident, theft, fraud, robbery, and snatching. For each category, we collected 600 legal cases, resulting in a total of 4,800 legal cases. (Note: The collected legal cases for each charge are from the courts in the same region and only include criminal first-instance cases.)

Define Legal Entities and Relationships. We have redefined 19 categories of legal entities and 11 categories of legal relationships to address the characteristics of cases involving ECC. By utilizing diverse entity and relationship information, we can accurately construct the criminal processes of legal cases and differentiate the subtle differences between ECC. The detailed legal entities and legal relationships are presented in Table 1.

We following the defined entities and relationships as presented in Table 1, with the annotation and verification of legal experts, we ultimately constructed a high-quality dataset for **Judicial Long** Text **Triple** Extraction (JLT). Detailed statistics regarding the dataset will be presented in Sect. 4.

1 https://wenshu.court.gov.cn/.

Table 1. Definition of legal entities and relationships

	Information						
Entity	Case	Defendant	Victim	Cause	Behavior	Tool	Injury
	Primary Culprit	Accessory	Recidivism	Surrender	Forgiveness	Amount	Total Amount
	Law	Charge	Prison Term	Penalty	Truthful Confession		
Relationship	Include	Involve	Crime Cause	Behavior Description	Use	Total	Injury Classification
	Crime Type	Sentencing	Violate	Judgment Information			

Defining Criminal Behavior Knowledge Graph. For predicting ECC tasks, the sequential information of criminal behaviors within criminal events is of paramount importance. Consequently, following the defined legal entities and relationships, we extracted the most relevant criminal behaviors of the defendant and their respective sequences. Criminal behaviors constitute KE, while sequence information constitutes CS. The extracted relevant entities include Crime Cause, Behavior, Tool, Injury Classification, and Charge. The extracted relevant relationships include Use, Cause, Favour, Subsequence, Lead to, and Constitute. From the JLT, we selected entities and relationships related to criminal behaviors and constructed CBKG using NetworkX[2]. At this stage, CBKG represents a large graph containing information about criminal behaviors. Subsequently, specific CS are extracted based on the behaviors corresponding to each legal case. The CBKG is illustrated in Fig. 3.

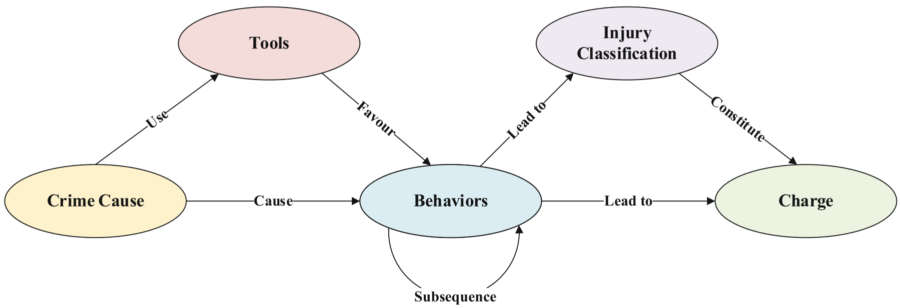

Fig. 3. An example of CBKG

3.3 Extract Module

In criminal cases in China, defendants often have a series of criminal behaviors that occur in a sequence from cause to outcome. This sequential information on criminal

[2] https://github.com/networkx/networkx.

behaviors can reflect the order in which the defendant engaged in criminal activities and provide additional perspectives for legal professionals in their judgment. Within a case, multiple behaviors of the defendant can be extracted, and then multiple paths related to these behaviors can be captured from the CBKG. By combining the captured paths, we can obtain a CS that reflects the differences in criminal behaviors between different charges, providing distinguishing features of the criminal process.

The Extract Module takes FD and CBKG as input data and is responsible for extracting KE and CS. Initially, all node information from CBKG is retrieved, and a string matching is conducted with the FD. Successfully matched nodes are identified as KE within the FD. Additionally, multiple paths related to behaviors are extracted from CBKG based on the sequence of node appearances, and these paths are combined to construct CS for criminal behaviors.

3.4 Encoder Module

The FD and KE both fall under textual information. In order to obtain rich features, we choose to encode them using a TextCNN. Specifically, the representation of the KE can provide the model with additional fine-grained semantic information.

For a legal case, denoted as $d_k = [w_1, w_2, \ldots, w_t, \ldots w_T]$, where T represents the number of words in the k-th case, and w_t represents the t-th word in the case facts d_k. For the KE extracted from the FD, denoted as $E_k = [e_1, e_2, \ldots, e_i, \ldots, e_I]$, where I represent the number of KE in the k-th case, and e_i represents the i-th KE in the KE sequence E_k. In this study, GloVe [13] embeddings are used to obtain word representations for each word in d_k and each element in E_k, represented as $\boldsymbol{w_t} \in \mathrm{R}^{\mathrm{m}}$ and $\boldsymbol{e_t} \in \mathrm{R}^{\mathrm{m}}$, respectively, where m is the dimensionality of the vectors. The representations of all the words in d_k are concatenated in the order they appear in the text, resulting in the representation $\boldsymbol{X}_{dk} = \boldsymbol{w_{1:T}} = [\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_t}, \ldots, \boldsymbol{w_T}]$ for the case facts. Similarly, the representations of all the elements in E_k are concatenated in the sequence order, resulting in the representation $\boldsymbol{X}_{Ek} = \boldsymbol{e_{1:T}} = [\boldsymbol{e_1}, \boldsymbol{e_2}, \ldots, \boldsymbol{e_i}, \ldots, \boldsymbol{e_I}]$ for the KE. These representations, $\boldsymbol{X}_{dk}$ and $\boldsymbol{X}_{Ek}$, are then inputted into the convolutional layer of the TextCNN. Different-sized convolutional kernels are used to extract text features. This process can be represented as Eqs. (1) and (2).

$$\boldsymbol{c} = f\left(\boldsymbol{W}_1 * \boldsymbol{X}_{d_k} + b_1\right) \tag{1}$$

$$\boldsymbol{l} = f\left(\boldsymbol{W}_2 * \boldsymbol{X}_{E_k} + b_2\right) \tag{2}$$

where $\boldsymbol{W_1}, \boldsymbol{W_2} \in \mathrm{R}^{\mathrm{h}\times\mathrm{m}}$ represent the sizes of the convolutional kernels, b_1, b_2 are biases, and $f(\cdot)$ represents a non-linear function. By using convolutional kernels of different sizes, feature sets are obtained for the case facts d_k and the KE sequence E_k, represented as $\boldsymbol{C} = [\boldsymbol{c_1}, \boldsymbol{c_2}, \ldots, \boldsymbol{c_{T\text{-}h+1}}]$ and $\boldsymbol{L} = [\boldsymbol{l_1}, \boldsymbol{l_2}, \ldots, \boldsymbol{l_{I\text{-}h+1}}]$, respectively. T and I represent the number of words in the FD and KE respectively. Finally, the maximum pooling is applied to obtain the text representation $\boldsymbol{C_{dk}}$ for the case facts d_k and the representation $\boldsymbol{L_{Ek}}$ for the KE E_k, capturing representative features. This process can be represented as Eqs. (3) and (4).

$$\boldsymbol{C_{d_k}} = [\max(\boldsymbol{C})] \tag{3}$$

$$L_{E_k} = [\max(L)] \tag{4}$$

3.5 Subgraph Encoder Module

Graph Convolutional Neural Networks (GCNs) typically aggregate neighboring node information using equal or pre-defined edge weights. In contrast, GAT can assign different weights to different nodes in the graph, allowing for varying importance levels of each neighboring node. Furthermore, GATs do not require the utilization of the entire graph; they only rely on first-order neighboring node information. This addresses some of the limitations of GCNs. Therefore, we choose GAT to encode the extracted CS.

Assuming that the CS contains N nodes, we denote the node vectors of graph as $\boldsymbol{h} = \{\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_N}\}, \boldsymbol{h_i} \in \boldsymbol{R^F}$ where F represents the feature dimension of input nodes. Next, the CS is fed into the GAT. When aggregating information between nodes, GAT incorporates an attention mechanism. The formula for calculating the attention coefficients e_{ij} is shown in Eq. (5).

$$e_{ij} = a\left(\boldsymbol{W}\boldsymbol{h_i}, \boldsymbol{W}\boldsymbol{h_j}\right) \tag{5}$$

where $\boldsymbol{W}$ represents the weight matrix that can be shared, and a denotes the shared attention mechanism. E_{ij} represents the relevance of node I to node j, where $j \in N_i$ and N_i represents all the first-order neighboring nodes of node i. Next, the attention obtained is normalized using the Softmax function. This normalization allows for easy comparison of attention weights between nodes. This process can be represented by Eq. (6).

$$\alpha_{ij} = softmax\left(e_{ij}\right) = \frac{\exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{W}\boldsymbol{h}_i \| \boldsymbol{W}\boldsymbol{h}_j\right]\right)\right)}{\sum_{k \in N_i} \exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{W}\boldsymbol{h}_i \| \boldsymbol{W}\boldsymbol{h}_k\right]\right)\right)} \tag{6}$$

where the attention mechanism is implemented as a single-layer feed-forward neural network with parameters denoted as $\boldsymbol{a}$. The activation function used is the LeakyReLU. $\boldsymbol{W}$ represents trainable parameters and || denotes the concatenation operation.

In order to stably represent nodes, we extend the attention mechanism to a multi-head attention mechanism, aiming to improve model stability. M is used to represent the number of attention heads. The final node representation is obtained by averaging the representations obtained through $\boldsymbol{M}$ attention heads. The output can be expressed as shown in Eq. (7).

$$\boldsymbol{h}_{i'} = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{j \in N_i}\alpha_{ij}^{M}\boldsymbol{W}^{M}\boldsymbol{h}_j\right) \tag{7}$$

where $\sigma(\cdot)$ represents the activation function. α_{ij}^{M} represents the value computed by the *m-th* attention head. $\boldsymbol{W^M}$ is the linear matrix used for the linear transformation of the input vector. $\boldsymbol{h}_i^{'}$ represents the feature obtained after feature extraction through the multi-head GAT, which captures the aggregated semantic information of neighboring nodes.

Next, applies max pooling to all the nodes $\boldsymbol{h}_i^{'}$ in the subgraph of the *k-th* case, resulting in a CS representation G_{d_k} that captures the sequential information of criminal behavior. This can be expressed as shown in Eq. (8).

$$G_{d_k} = \text{MaxPooling}(\boldsymbol{h}') \tag{8}$$

3.6 Feature Fusion

The KE of the case, as well as the relevant CS representation, can provide additional information for representing the case facts. Therefore, in this paper, the CS representation $\boldsymbol{G_{dk}}$, KE representation $\boldsymbol{L_{Ek}}$, and FD representation $\boldsymbol{Cd_k}$ are concatenate and fed into a fully connected layer, resulting in a representation denoted as $\boldsymbol{p}$, as shown in Eq. (9). Finally, this representation is inputted into a softmax to predict the involved charges. This process is represented by Eq. (10).

$$\boldsymbol{p} = concat\left[\boldsymbol{L_{E_k}}, \boldsymbol{C_{d_k}}, \boldsymbol{G_{d_k}}\right] \tag{9}$$

$$z = softmax(\boldsymbol{p}) \tag{10}$$

4 Experimental Setup

4.1 Dataset

This study conducted experiments on two charge prediction datasets consisting of real legal cases, JLT and CAIL-8. The JLT includes 8 ECC and comprises 3,842 cases. The CAIL-2018 is an official charge prediction task dataset released by CAIL and contains 202 charges. The CAIL-8 dataset is a subset of CAIL-2018, consisting of legal cases with the same charges as the JLT dataset. (Note: Extracting legal cases with the same charges as the JLT dataset aims to fully validate the effectiveness of the CBKG. In the experiments, CS was extracted from the CBKG, while KE and FD corresponded to the content of the respective dataset.) We conducted a detailed analysis of the case length, and the statistical data is presented in Table 2.

4.2 Implementation Details

The baseline and CP-KG were trained and tested on NVIDIA Tesla V100. According to the analysis in Table 2, the average length of a legal case fact description when using characters as the smallest semantic unit is 497, while it is 371 when using words as the smallest semantic unit. Therefore, in the experiments, the fixed length for input text sequences was set to 400, and the fixed length for KE was set to 50. The experiments utilized word embeddings trained by the GloVe with a dimension of 200 for parameter settings. The convolutional kernel sizes were set to 2, 3, 4, and 5. The training was performed for 20 epochs, and the Adam optimizer was used. The dropout rate was set to 0.5, and the learning rate was set to 1e−3.

Table 2. Dataset statistics information

	JLT-Train	JLT-Test	JLT-Valid	CAIL-8-Train	CAIL-8-Test	CAIL-8-Valid
Total cases	2304	769	769	8243	5227	2441
Average words	391.08	398.76	396.08	281.00	263.48	269.99
Average characters	530.87	564.76	555.82	352.99	331.30	340.56

4.3 Metric

The performance of the CP-KG in this experiment was evaluated using Accuracy (Acc), Macro-Precision (Mac-P), Macro-Recall (Mac-R), and Macro-F1 (Mac-F1) as metrics.

4.4 Baseline Methods

To evaluate the experimental performance of the CP-KG, the following methods were chosen as baseline. All methods used the default settings from the original papers.

- **TFIDF-SVM** [14] is a baseline model provided in the CAIL2018 competition. It uses TF-IDF to extract text features and employs a Support Vector Machine (SVM) for case fact classification.
- **TextCNN** [15] utilizes multiple convolutional layers with different kernel sizes followed by max pooling to encode case facts and predict charges.
- **Bi-GRU** employs a Bi-directional Gated Recurrent Unit to capture text features.
- **TOPJUDGE** [3] is a topological multitask framework that captures the topological dependencies among three judgment prediction subtasks: charge prediction, law article recommendation, and sentence prediction. It uses a Directed Acyclic Graph (DAG) structure.
- **NeurJudge** [16] primarily uses the Bi-GRU approach to encode texts and construct charge graphs and legal provision graphs using charge definition and legal provision. It utilizes graph decomposition to distinguish confusing legal provisions and charge categories and combines the obtained label semantic information with case facts for prediction.

5 Experimental Results

5.1 Analyze

From the experimental results in Table 3, it can be observed that the results of deep learning are generally higher than those of the TFIDF-SVM. This is because the length of case facts is not uniform, and there is a considerable amount of content in the FD. The use of the Bi-GRU leads to the issue of content forgetting, resulting in poor performance in charge prediction. TextCNN effectively captures local features of the text, thus demonstrating better performance in charge prediction. TOPJUDGE, also based on a CNN, differs in associating the three judgment prediction subtasks, leading to superior performance. NeurJudge is based on the Bi-GRU and also establishes some degree of correlation in the three subtasks. However, the graph decomposition operation in this method is more suitable for multiple classes of charges, making it difficult to effectively distinguish distinctive features and resulting in lower prediction performance than other models. The proposed CP-KG combines CS, KE, and FD for feature fusion and prediction. Experimental results demonstrate that CP-KG achieves State-Of-The-Art (SOTA) performance among the baseline. This is primarily due to the fine-grained semantic information provided by the KE in the cases, as well as the criminal process of the defendant and the relationships between elements provided by the CS, which enable a deep understanding of the FD.

Table 3. Comparison of experimental results

Dataset	JLT				CAIL-8			
Methods	Acc	Mac-P	Mac-R	Mac-F1	Acc	Mac-P	Mac-R	Mac-F1
TFIDF-SVM	0.7757	0.7722	0.7749	0.7331	0.8041	0.8191	0.8265	0.7641
TextCNN	0.9831	0.9813	0.9784	0.9796	0.9346	0.9044	0.8782	0.8773
Bi-GRU	0.8388	0.8145	0.8142	0.8090	0.9038	0.8611	0.8791	0.8569
TOPJUDGE	0.9863	0.9897	0.9784	0.9839	0.8719	0.8581	0.8608	0.8334
NeurJudge	0.9714	0.9697	0.9672	0.9683	0.9066	0.8644	0.8967	0.8699
CP-KG	**0.9922**	**0.9917**	**0.9903**	**0.9910**	**0.9474**	**0.9165**	**0.8975**	**0.9023**

5.2 Ablation Study

We conducted an extensive ablation study on CP-KG to validate the effectiveness of the KE and CS. In these experiments, the names of methods indicate the encoder and the encoded object.

From Table 4, it can be observed that incorporating the KE and CS along with the FD leads to improved prediction performance compared to using only the FD, KE, or CS. Mac-F1 increases by an average of 2%. Furthermore, the results show that incorporating the CS yields higher performance than incorporating the KE. This indicates that the sequential information of the elements in the CBKG provides additional semantics, which helps differentiate between ECC. The ablation study results on the CAIL-8 demonstrate that the CBKG is applicable not only to the JLT but also to different cases with the same charges. This validates the generalizability of the CBKG and the effectiveness of the CP-KG.

Table 4. Ablation study results (For example, TextCNN w/KE means that TextCNN was used to encode the Key Elements).

Dataset	JLT				CAIL-8			
Methods	Acc	Mac-P	Mac-R	Mac-F1	Acc	Mac-P	Mac-R	Mac-F1
TextCNN w/FD	0.9831	0.9813	0.9784	0.9796	0.9346	0.9044	0.8782	0.8773
TextCNN w/KE	0.9727	0.9709	0.9727	0.9716	0.9045	0.8545	0.8839	0.8603
GAT w/CS	0.9493	0.9503	0.9432	0.9458	0.9013	0.8505	0.8676	0.8547
TextCNN w/FD+KE	0.9896	0.9879	0.9876	0.9877	0.9334	0.8976	0.8865	0.8821
TextCNN w/FD+CS	0.9909	0.9899	0.9874	0.9885	0.9336	0.8966	0.8903	0.8867
CP-KG	**0.9922**	**0.9917**	**0.9903**	**0.9910**	**0.9474**	**0.9165**	**0.8975**	**0.9023**

6 Conclusion

In this paper, we propose the CP-KG model that integrates a Criminal Behavior Knowledge Graph (CBKG) to address the problem of overlooking crucial elements and sequential information in legal cases in the context of charge prediction methods. The CP-KG first extracts Key Elements (KE) and Case Subgraphs (CS) from the CBKG and the Fact Descriptions (FD). Subsequently, these KE and CS are encoded by a combination method of TextCNN and GAT. Finally, KE, CS, and FD representations are fused to predict the defendant's charges. Experimental results demonstrate that CP-KG outperforms the baseline and achieves state-of-the-art performance, with 25.79% and 13.82% improvements in the Macro-F1 metric. Moreover, the ablation study validates the effectiveness of KE and CS, as they collectively enhance the predictive performance of the CP-KG. Additionally, the CBKF constructed in this paper provides fine-grained semantic information for charge prediction and exhibits generalizability, making it transferable to related legal tasks.

Acknowledgments. This paper was supported by the National Natural Science Foundation of China (12204062, 61806103, 61562068), National Natural Science Foundation of Inner Mongolia, China (2022LHMS06001), Basic Scientific Research Business Project of Inner Mongolia Normal University (2022JBQN106, 2022JBQN111).

References

1. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
2. Jiang, X., Ye, H., Luo, Z., Chao, W., Ma, W.: Interpretable rationale augmented charge prediction system. In: Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, pp. 146–151 (2018)
3. Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., Sun, M.: Legal judgment prediction via topological learning. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3540–3549 (2018)

4. Yang, W., Jia, W., Zhou, X., Luo, Y.: Legal judgment prediction via multi-perspective bi-feedback network. In: Twenty-Eighth International Joint Conference on Artificial Intelligence (2019)
5. Zhao, J., Guan, Z., Xu, C., Zhao, W., Chen, E.: Charge prediction by constitutive elements matching of crimes. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, vol. 22(23–29), pp. 4517–4523 (2022)
6. Chen, S., Wang, P., Fang, W., Deng, X., Zhang, F.: Learning to predict charges for judgment with legal graph. In: Artificial Neural Networks and Machine Learning, pp. 240–252 (2019)
7. Chen, J.X., Huang, Y.J., Cao, G.J., Yang, F., Li, C., Ma, Z.B.: Research and implementation of judicial case visualization based on knowledge graph. J. Hubei Univ. Technol. **34**(05), 72–77 (2019)
8. Chen, W.Z.: Research on Legal Text Representation Method Fused with Knowledge Graph. GuiZhou University (2020)
9. Guo, J.: Research and Implementation of Auxiliary Judgment Technology Based on Affair Graph. Beijing University of Posts and Telecommunications (2021)
10. Chen, Y.G.: Research on Entity Relationship Extraction Algorithm for Legal Documents. Dalian University of Technology (2021)
11. Hong, W.X., Hu, Z.Q., Weng, Y., Zhang, H., Wang, Z., Guo, Z.X.: Automatic construction of case knowledge graph for judicial cases. J. Chin. Inf. Process. **34**(01), 34–44 (2020)
12. Wang, Z.Z., et al.: Sentencing prediction based on multi-view knowledge graph embedding. Pattern Recogn. Artif. Intell. **34**(07), 655–665 (2021)
13. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
14. Xiao, C., Zhong, H., Guo, Z., et al.: CAIL2018: a large-scale legal dataset for judgment prediction. arXiv preprint (2018)
15. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint (2014)
16. Yue, L., Liu, Q., Jin, B., et al.: NeurJudge: a circumstance-aware neural framework for legal judgment prediction. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 973–982 (2021)
17. Ma, L., et al.: Legal judgment prediction with multi-stage case representation learning in the real court setting. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 993–1002 (2021)
18. Lyu, Y., et al.: Improving legal judgment prediction through reinforced criminal element extraction. Inf. Process. Manage. **59**(1), 102780 (2022)
19. Dong, Q., Niu, S.: Legal judgment prediction via relational learning. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 983–992 (2021)
20. Feng, Y., Li, C., Ng, V.: Legal judgment prediction via event extraction with constraints. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, pp. 648–664 (2022)