Information Processing and Management 60 (2023) 103421

Contents lists available at ScienceDirect

Information Processing and Management

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

Criminal Action Graph: A semantic representation model of judgement documents for legal charge prediction

Geya Feng, Yongbin Qin *, Ruizhang Huang, Yanping Chen

Text Computing & Cognitive Intelligence Engineering Research Center of National Education Ministry, College of Computer Science and Technology, Guizhou University, Guiyang, 550025, PR China
State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, 550025, PR China

ARTICLE INFO

Keywords:
Data mining
Graph representation
Semantic information
Judgement document

ABSTRACT

Semantic information in judgement documents has been an important source in Artificial Intelligence and Law. Sequential representation is the traditional structure for analyzing judgement documents and supporting the legal charge prediction task. The main problem is that it is not effective to represent the criminal semantic information. In this paper, to represent and verify the criminal semantic information such as multi-linked legal features, we propose a novel criminal semantic representation model, which constructs the **C**riminal **A**ction **G**raph (*CAG*) by extracting criminal actions linked in two temporal relationships. Based on the *CAG*, a Graph Convolutional Network is also adopted as the predictor for legal charge prediction. We evaluate the validity of *CAG* on the confusing charges which composed of 32,000 judgement documents on five confusing charge sets. The *CAG* reaches about 88% accuracy averagely, more than 3% over the compared model. The experimental standard deviation also show the stability of our model, which is about 0.0032 on average, nearly 0. The results show the effectiveness of our model for representing and using the semantic information in judgement documents.

1. Introduction

Legal charge prediction (LCP) is an important task in Artificial Intelligence and Law. The task predicts the charge of new cases by analyzing historical information from judgement documents. It is helpful to assist judges in their decisions. Recently, researchers have found that confusing cases limit the prediction accuracy in LCP. Confusing cases refer to the cases that have similar or even the same criminal actions and criminal tools but have been sentenced to different charges.

To solve this problem, researchers try to use semantic information in judgement documents, e.g., discriminative legal attributes (Hu, Li, Tu, Liu, & Sun, 2018), casual relationships between legal features (Liu, Yin, Feng, Wu, & Zhao, 2021), or structural relationships between judgement documents and legal entities (Bi, Ali, Wang, Wu, & Qi, 2022). However, the performance is not as expected because they can not extract association information between features, where the prediction noise is included and a sequential representation could not capture the differential information between judgement documents. One way to resolve these problems is to use as many legal features as possible to represent judgement documents that contain semantic information. It means that it should focus on other more influential legal features in judgement documents for LCP.

* Corresponding author at: State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, 550025, PR China .

E-mail addresses: gs.fenggy20@gzu.edu.cn (G. Feng), ybqin@gzu.edu.cn (Y. Qin), cse.rzhuang@gzu.edu.cn (R. Huang), ypench@gzu.edu.cn (Y. Chen).

https://doi.org/10.1016/j.ipm.2023.103421
Received 16 March 2023; Received in revised form 18 May 2023; Accepted 31 May 2023
Available online 21 June 2023

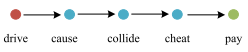
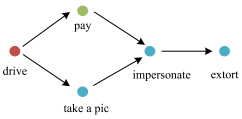
Case description	CAS	CAG	Charge result
...On July 22, 2013, the defendant Huang **drove** a car at the Y district Z of the city X, and **caused** a traffic accident by deliberately **colliding** with the small truck driven by the victim Yuan, **cheating** Yuan Moumou to **pay** the compensation in RMB C Yuan...	[drive, cause, collide, cheat, pay]		**Fraud**
...At about 3 pm on May 20, 2015, defendants Wang and Li took Yang's "black taxi" near X District to Y Town, at a price of A yuan. When the car **drove** near the Z College, the defendant Wang **took a picture** of his mobile phone ***while*** Li **paid** Yang, **impersonated** to be the staff of the Transportation Management Office to enforce the law, and asked Yang Mou to drive the car to M County . The transportation management office will deal with it and pay a fine of B yuan. Later, the defendants Wang and Li used this as the name to **extort** Yang for C yuan...	[drive, pay, *while*, take a pic, impersonate, extort]		**Extortion**

Fig. 1. This is a real example of confusing cases and their difference in criminal actions. The case descriptions are translated into English for reading easily. In these two real-world cases, they have some criminal actions that are the same, such as *drive* and *pay*, but they are charged with different charge results. However, the Criminal Action Graph (*CAG*) can help to distinguish confusing cases since the *CAG* consists of not only the criminal actions but also their temporal relationships.

1.1. Research objectives

Criminal actions and temporal relationships between them: In Chinese Criminal Law, the criminal action is an important component of special constitutive elements of a crime. But depending on only criminal actions cannot distinguish the confusing cases independently. Therefore, it is essential for LCP to consider logical relationships between the actions. In judgement documents, the temporal relationships between criminal actions are also important, especially in confusing cases. For example, *extortion* and *fraud* cases have similar criminal actions, such as inflicting violence on the victim and taking away the victim's belongings. It is difficult to distinguish *extortion* and *fraud* cases only by using criminal actions, because they contains similar actions. Furthermore, the temporal relationships can help to distinguish these confusing cases. For example, when the two criminal actions occur simultaneously, the charge result often given is the *fraud* accusation. On the contrary, the *extortion* cases always show the sequential occurrence of these two criminal actions, like the suspect threatening the victim with verbal violence and then handing over his money (ZiXian, 2021). This simple sample shows how the temporal relationship between comparable criminal events helps judges to differentiate confusing cases. Likewise, as is shown in Fig. 1, in the real-world dataset we used, temporal relationships between actions could apply to identify confusing cases when they have similar or even the same criminal actions.

The graph representation: To correctly present the temporal relationships in the judgement document, we choose the graph structure that comprises nodes and edges since edges can exhibit the relationship between nodes well. In a similar vein, the Knowledge Graph (KG) has changed the typical way of the presentation of field knowledge. It has expressed knowledge as a graph with entities and their association, transforming unstructured data such as text into structured data. Currently, the latest research work focuses on the Temporal Knowledge Graph (TKG) (Jin, Qu, Jin, & Ren, 2019; Jung, Jung, & Kang, 2021; Zhu, Chen, Fan, Cheng, & Zhang, 2021), which includes time information beyond typical KG. In the legal domain, judgement documents are perfectly suitable data for temporal analysis as a form of tightly formatted text, but nobody has joined temporal graph theory into the field of judgement document representation.

1.2. Research method

In this paper, we propose a *CAG* representation. It is a new way to represent judgement documents based on a graph structure that could store multi-linked criminal actions and their relationships. Comparing with sequential representation in the related work, our method contains *concurrency* temporal relationship to describe judgement documents more accurately and cover more semantic information. Our model consists of two modules: a construction module and a prediction module. The construction module builds up the *CAG* with extracted criminal actions and learned temporal relationships. To distinguish the different temporal relationships mentioned above, we purpose learning rules when the module scans the criminal action sequence. After constructing *CAG* completely, the prediction module predicts the charge of judgement document with the SAGPool (Lee, Lee, & Kang, 2019) as the predictor to classify confusing cases. Experimental results on five confusing charges sets demonstrate the validity of the *CAG*. In summary, our major contributions are as follows:

- we propose *CAG*, a novel semantic representation model base on the graph to represent the semantic information of the case description in judgement documents, which can use and store both legal features and their relationships features;
- we propose an algorithm that can transfer the judgement document to *CAG* automatically, thus eliminating the need to tag features manually;
- compared to related works on the analysis in judgement document about the causal relationship or the structural link between legal features, we focus on the temporal relationship that has acquired less attention but is significant.

2. Related works

Related work about our model are roughly divided into four parts: (1) Legal charge prediction, (2) Graph structure in legal textual analysis,(3) temporal relationships between features and their advantages, (4) the selection and comparison of a deep learning model for confusing charge prediction. Each part is presented as follows.

2.1. Legal charge prediction

Legal judgement analysis gets worldwide attention in the legal domain. Researchers from different countries have proposed differential algorithms for analyzing legal documents in their countries. As in the Indian legal texts, Bi-LSTM is used in classifying multi-label texts (Vaissnave & Deepalakshmi, 2022). A heterogeneous graph is applied to identify legal statute of legal judgement (Paul, Goyal, & Ghosh, 2022). Also in Brazil, multi-step word embedding algorithm (Coelho et al., 2022) and topic modeling (Aguiar, Silveira, Furtado, Pinheiro, & Neto, 2022) take advantage of legal charge prediction. Random forest algorithm (Chen, Wu, Chen, Lu, & Ding, 2022), label-attention and domain-specific pre-training are utilized in the US legal documents. Legal document analysis in the EU focuses on the construction of a legal knowledge service platform (Schneider et al., 2022) and legal knowledge network (Sulis et al., 2022). Although in countries such as Cuba, Artificial Intelligence and Law is a future project on the agenda (Rodríguez Rodríguez, Amoroso Fernández, Peña Abreu, & Sergeevich Zuev, 2021). Therefore, legal text analysis is an important area of global interest.

When Chinese LCP was first proposed in 2018 (Xiao et al., 2018), it was one of the important components of legal judgement prediction. Researchers used the machine learning method in LCP such as SECaps (He, Peng, Le, He, & Zhu, 2019) or mixed model (Wenguan, Yunwen, Hua, Yanneng, & Huiyu, 2019). Wang et al. presented a model based on the hybrid attention and CNN model, which combines the improved hierarchical attention network (iHAN) and the deep pyramid convolutional neural network (DPCNN) by ResNet (Wenguan et al., 2019). However, it failed to make better legal charge prediction in confusing cases. LCP was regarded as a textual classification task in the legal domain at first, which used typical classification algorithms such as TextCNN (Sun, Ma, Ni, & Bian, 2018). It usually suffer from low performance because it is difficult to distinguish confusing cases. In order to rectify this problem, researchers have mined more information of judgement documents or expanded the documents with legal concepts, for example, legal statutes (Bi et al., 2022) or data augmentation (Csányi & Orosz, 2022). There are a few works for applying semantic information in the legal judgement document, while many researchers are focusing on the textual structure of legal texts. Both CEEN (Lyu et al., 2022) and TOPJUDGE (Zhong et al., 2018) show good performance in the joint legal prediction task. CEEN achieves mutual tuning of multi-task predictions (i.e. law articles prediction, charge prediction, and term of penalty prediction) by extracting multiple elements of the case. On the other side, TOPJUDGE achieves improved accuracy by linking different multi-task prediction topologies in a joint tuning approach.

2.2. Graph structure in legal textual analysis

The application of deep learning in natural language processing starts with sequence representation, which was applied to deal with the textual serialization as TextCNN (Sun et al., 2018). Researchers expanded serialization to textual graphs within various associations like syntax-based sentence constituent relations (Bastings, Titov, Aziz, Marcheggiani, & Sima'an, 2017) or co-reference relationships between recognized entities (Song et al., 2018). The knowledge graph (KG) has flourished over the decade since it was first proposed and consists of entities and their linking learning from several sorts of texts (Chen, Jia, & Xiang, 2020). Recently, an event logic graph (ELG) has been proposed that can reveal evolutionary patterns and development logics of real-world events (Ding, Li, Liu, & Liao, 2019). In the legal domain, the heterogeneous graph has been used to compare the similarities between judgement documents (Bi et al., 2022; Paul et al., 2022). However, no study to date has examined the graph representation applied in legal charge prediction, while other studies have used sequence representation or outer legal knowledge as usual.

2.3. Temporal relationship used between textual features

In recent works, the sequence is always used to express time linking among events in texts and fit some statistical process like oriented Dirichlet Process (Erfanian, Cami, & Hassanpour, 2022) or Hwakes Process (Zhang, Liang, Sheng, & Shao, 2022). The TimeBank, an event corpus, proposed thirteen types of temporal relationships (Pustejovsky et al., 2003). The inner logical time relationships between features are always ignored. To store more temporal information between criminal actions, we expand the time sequence to *CAG* within the sequential relationship as well as concurrency.

2.4. Model selection and comparison

The current researches on analyzing judgement documents mainly uses the deep learning thesis. They take the case description as input data and the corresponding charge as the text label. Then, the legal charge prediction task is converted into a text classification problem (Qin, Huang, & Luo, 2022). Liu et al. exposed a novel sequence-based Causal Inference (GCI) framework, which built causal chains from fact descriptions (Liu et al., 2021). This model includes more semantic information than other models base on deep learning. But the casual connections only came from the co-occurrent frequency between features that display poor interpretability. Since CAG mainly expresses legal text data semantically, we also compared the current more advanced text semantic expression

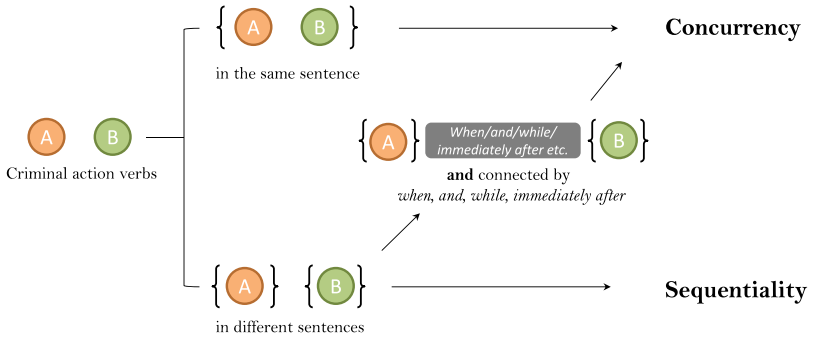


Fig. 2. This figure shows the process of how to learn the temporal relationships between criminal actions in judgement document.

models: Bert (Devlin, Chang, Lee, & Toutanova, 2018), RoBERTa (Liu et al., 2019) and Xlnet (Yang et al., 2019). These pre-trained models re-encode the text through adaptive and auto-regressive methods to form text expressions containing semantic information. However, the application of confusing cases, which is the challenge of legal charge prediction, provides a valuable way for analyzing judgement documents.

3. Framework

Before describing our method for legal charge prediction, we first provide a formalized discussion about the data structure our model. The legal charge prediction of the confusing case task is defined as: given criminal facts in judgement documents, the task is to classify these facts into the *result* of right charge from a confusing charge set $r = \{r_1, r_2, r_3, \ldots, r_m\}$. In every charge, there are two or three results, which means the confusing crime names of every fact. In this paper, m is defined as 2 in binary classification and 3 in multi-classification.

Definition 1 (*CAS*)**.** Given a CAS (Criminal Action Sequence) $S = \{v_1, v_2, v_3, t, v_4, t, v_5, \ldots, v_n\}$, the sequence describes the main feature in a judgement document denoted as $v_n \epsilon \mathbb{V}$, while $\mathbb{V}$ is the set of verbs in all the texts, and t is the concurrent conjunction such that two actions connected by t occur simultaneously. In this paper, we assume that only two actions can happen at the same time. For example, in S, v_4 and v_5 are the meantime verbs while v_1 and v_2 are the sequential verbs. An example is given in Fig. 1. The CAS of Fraud case is {drive, cause, collide, cheat, pay}. The CAS of Extortion case is {drive, take a pic, *while*, pay, impersonate, extort}. However, the construction of CAG is based on scanning CAS.

Definition 2 (*Temporal Relationship*)**.** As in Fig. 2, temporal relationships have been delivered into two pieces, *concurrency* and *sequentiality*. In the first step, the textual positional relationship of two verb words should be identified, if they are in the same sentence. It means that they could be determined into *concurrency* temporal relationship. On the other hand, if they are in diverse sentences, they would be determined into *sequentiality* link. In addition to the concerned positional relation, the timestamps could tag the *concurrency* relation between verbs. Therefore, if two criminal action verbs are in different sentences, they will be established into *sequentiality* as usual, but if the sentences are joined by a concurrent word, our method will learn the *concurrency* association among them.

Definition 3 (*CAG*)**.** A criminal action temporal graph (CAG) is defined as a directed graph with timestamps $G = (V, R, T)$, where V and R are the verb set and relation set, respectively. T is the set of valid temporal concurrent words. Each graph presentation of the judgement document in *CAG* is represented as an adjacency matrix, which has a diverse topological structure within each text. As shown in Fig. 3, the common structures include both cycle and no cycle.

As shown in Fig. 3, our method consists of two components, construction and prediction. In the construction module, it extracts CAS and learns temporal relationships, then constructs *CAG*s. In the next module, it uses *CAG*s to predict charge results.

3.1. Construction module

In this module, the target of output is *CAG*, which consists of criminal actions and temporal relationships between them. As the definition mentioned above, to construct the *CAG*, the construction module should get two types of features, criminal action verbs and timestamps. In our model, the Chinese text segmentation tool *jieba* is adopted for speech tagging. v is used to represent criminal actions in a criminal case description, while t represents timestamps, i.e., *when*, *and*, *while*, *immediately after*, *suddenly*, *a few minutes*.

A temporal relationship rule base is used to support the construction module. Details of the rule base are shown in Fig. 4. In this paper, two temporal relationships are included: the sequentiality relationship and the concurrency. In our rule base, these two temporal relationships among criminal actions will combine four situations, which are the basic elements of *CAG*s. *CAG*s are made with the arrangements of temporal connections.

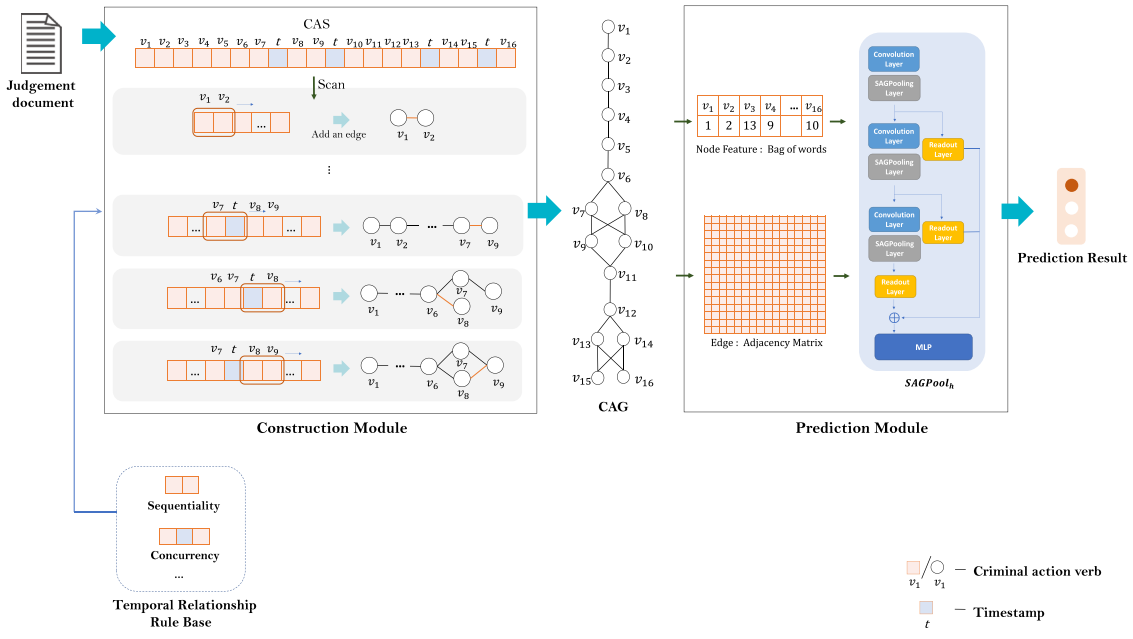


Fig. 3. This figure exhibits the framework of our approach, which consists of two modules, construction and prediction. First, we extract the judgement document into CAS and then learn the temporal relationships among extracted features to construct the *CAG* with criminal actions and learned relationships. The learning rule of temporal relationships is shown in Fig. 4. Afterward, we choose a suitable GCN classifier as the predictor for the *CAG* to distinguish confusing cases.

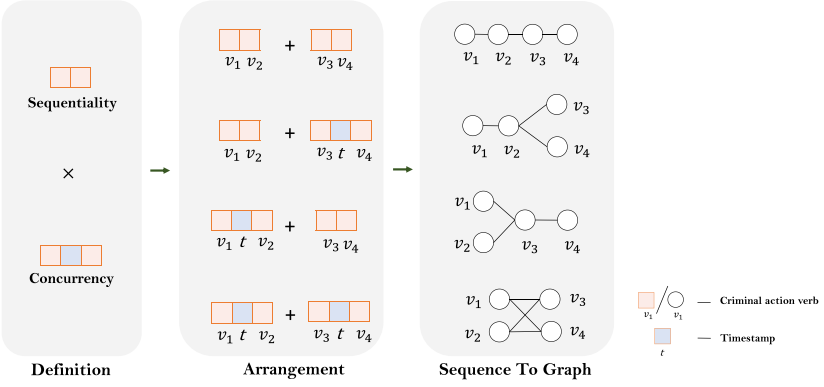


Fig. 4. This figure exhibits rules of the temporal relationship rule base in the construction module, which helps the model to transfer CAS to *CAG* when the module only focuses on two temporal relationships, sequentiality and concurrency. In the model, two temporal relationships are defined, which can combine four arrangements. In our model, we define that only two criminal actions are concurrent.

To construct *CAG*s, the most crucial problem is how to describe the temporal relationship between crime actions in the judgement document. We define the types of temporal relationships and recount how to distinguish the temporal relationships between criminal action verbs in the judgement document. They are discussed as follows.

Temporal relationship definition. The TimeBank is an event corpus that was put forward in 2002 and described temporal relationships between events in each sentence of the corpus (Pustejovsky et al., 2003). The corpus has represented 13 of types temporal relationships between two events, including *before*, *after*, *includes*, *is included*, *holds*, *simultaneous*, *immediately after*, *immediately before*, *identity*, *begins*, *ends*, *begun by*, *ended by* (Pustejovsky et al., 2003). Combined with reality, the passive voice is not included in Chinese grammar, which we will not think about in the temporal relationship. Moreover, judgement documents are written in chronological order, which means that the time relationship between verbs in Chinese grammar includes two types, which are *concurrency* and *sequentiality*. The *concurrency* relationship between verbs means two verbs of actions concur at the same time, while the *sequentiality* relationship means the latter action of the verb happens after the former one.

Temporal relation distinguish. After we define the types of temporal relationships between criminal action verbs in the judgement document, the next step is to distinguish the relationship between the two verbs. In the previous part, we have defined two types of relationships between crime verbs. Since the judgement documents are written in chronological order, the *sequentiality*

Algorithm 1: The construction module

Input: CAS $S = \{v_1, v_2, v_3, t, v_4, t, v_5, \cdots, v_m\}$.
Output: *CAG*'s adjacency matrix A, *CAG*'s word bag of criminal actions V.
Build up the adjacency matrix:
while $i \neq m$ **do**
  Learn temporal relationship *concurrency* or *sequentiality*:
  if v_i, v_{i+1} *are connected* **then**
    the temporal relationship between v_i and v_{i+1} is *sequentiality*, and note an edge between node v_i and v_{i+1} in adjacency matrix A;
  end
  if v_i, t *are connected* **then**
    the temporal relationship between v_i and v_{i+1} is *concurrency*, and note an edge between node v_i and v_{i+2} in adjacency matrix A;
  end
  if t, v_i *are connected* **then**
    the temporal relationship between v_i and v_{i+1} is *concurrency*, and note an edge between node v_i and v_{i-2} in adjacency matrix A;
  end
  $i++$;
end
Tag the criminal actions:
Scan all the criminal actions in the dataset, put all the non-repeating actions into one word array;

relationship will be distinguished clearly when the behavior of verbs in different sentences is separated by commas. To distinguish the *concurrency* relationship, we will discuss it in two cases. When the two verbs are in the same sentence, we believe that the two behaviors occur simultaneously. Especially when the two verbs are in different sentences, but there is a time adverb, which means the two actions are occurring at the same time, such as *when*, *and*, *while*, *immediately after*, then we will consider that the relationship between these two actions is *concurrency* relationship. Fig. 2 shows this process.

To structure the *CAG*, we should learn the temporal relationship among verbs from the CAS. In this part, we have written an algorithm to convert CAS to a matrix of *CAG* automatically. As Algorithm 1 shows, we can regard the adjacent verbs that connect by concurrent word as a whole, which builds up the *sequentiality* relationship with the former verb.

3.2. Prediction module

Through the above step, we will get an undirected graph to describe the fact of the criminal case, in which nodes constructed by the action verbs and edges represent temporal relationships between adjacent verbs. We use only one type of node and only one kind of edge to represent the temporal relationship between the defendant's actions in the criminal case from the judgement document.

According to the graph data structure, it is fast, convenient, and mainstream to use a graph neural network (GNN) to classify different graphs. Considering many different types of topological structures in our *CAG*, we choose the SAGPool, which fuses the self-attention mechanism and three convolution layers into a graph neural network (Lee et al., 2019). The SAGPool has two architectures, one for a small dataset while another is for the larger size dataset, so we choose the large one called the global pooling architecture as $SAGPool_h$.

Convolution layer. The model uses the classical definition of the graph convolution layer. Eq. (1) has shown the calculation process of the convolution layer.

$$h^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}h^{(l)}\Theta) \tag{1}$$

$$h^{(l+1)} = ReLU(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}h^{(l)}\Theta) \tag{2}$$

Where the $h^{(l)}$ means the node representation of lth layer and $\Theta \epsilon \mathbb{R}^{F \times F'}$ means the weight of the convolution while F is the input feature dimension and F' is the output feature dimension. σ is the activation function, which in our paper is the Rectified Linear Unit (ReLU), so Equation (1) can be expressed more accurately as Equation((2)). $\widetilde{A} \epsilon \mathbb{R}^{N \times N}$ means the adjacency matrix with self-connections of the input graph, and $\widetilde{D} \epsilon \mathbb{R}^{N \times N}$ is the degree matrix of $\widetilde{A}$.

SAGPooling layer. In this layer, two principal elements will be used. The first part is the self-attention mask. The model will use the graph convolution to calculate the self-attention scores as in the Equation ((3) or (4)) (Kipf & Welling, 2016).

$$Z = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}X\Theta_{att}) \tag{3}$$

$$Z = tanh(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}X\Theta_{att}) \tag{4}$$

Table 1
Overview of dataset.

Charge Sets	Charges Of Set	#Doc count	#Doc length avg	#Word avg	#Features avg
F&E	Fraud	3536	713	405	81
	Extortion	2149	466	269	54
AP&DD	Abuse of Power	1950	1016	569	123
	Dereliction of Duty	1938	780	438	102
E&MPF	Embezzlement	2391	1276	716	145
	Misappropriation of Public Funds	1998	576	314	66
Violent Acquisition	Robbery	5020	743	435	82
	Seizure	2113	463	265	48
	Kidnapping	622	692	414	83
Personal Injury	Intentional Injury	6377	656	391	79
	Murder	2282	665	404	87
	Involuntary Manslaughter	1989	365	209	48

$\widetilde{D}$ and $\widetilde{A}$ are the same means as in the Equation ((1) or (2)), and $X \in \mathbb{R}^{N\times F}$ is the N node and F dimensional feature of the input graph, while $\Theta_{att}\epsilon\mathbb{R}^{F\times 1}$ is the only parameter of the SAGPooling layer. Utilizing the attention score of the pooling layer, the result will be based on both features and topology. Then we need to select the saved node idx (Gao & Ji, 2019), which is calculated by the top-rank function that depends on the value of Z and the top $\lceil kN \rceil$ node.

$$idx = top - rank(Z, \lceil kN \rceil) \tag{5}$$

$$X' = X_{idx,:}\ ,\quad X_{out} = X' \odot Z_{rank},\quad A_{out} = A_{idx,idx} \tag{6}$$

where $X_{idx,:}$ is the row-wise matrix of indexed feature, $\odot$ is the broadcasted elementwise product, and $A_{idx,idx}$ is the row-wise and col-wise indexed adjacency matrix. X_{out} and A_{out} are the new feature matrix and the corresponding adjacency matrix.

Readout layer. In the model, a readout layer is worked as aggregating features of the node to make a fixed size representation, which is the calculated output feature of this layer as follows:

$$s = \frac{1}{N}\sum_{i=1}^{N} x_i \| \max_{i=1}^{N} x_i \tag{7}$$

where x_i is the feature vector of ith node, and $\|$ denotes concatenation.

MLP & Loss. The MLP layer integrates three fully connected layers, uses the ReLU function for activation, and calculates and processes the outputs from the three global pooling layers of SAGPool, and the output($logits$) is the predicted probability of case classification. In our model, the cross-entropy loss function is used as the loss function, and in order to reduce the amount of calculation, only the training data($train\_label$) is used to calculate the model loss. The calculation formula is as follows:

$$\mathcal{L} = -log(\frac{exp(logits[train\_label])}{\sum_i exp(logits[i])}) \tag{8}$$

4. Experiments and evaluation

4.1. Dataset

To compare with the causal model GCI, our experiments use the same dataset source from the Chinese AI and Law Challenge (CAIL2018) (Xiao et al., 2018), which is divided into five groups where each is named as confusing cases sets (House, 2017). We used two parts of the judgement document, including the case description and the criminal charge. The former will be processed into *CAG* and the latter will tag the case description. The overview of the dataset is shown in Table 1. It consists of the count of each charge, the length average of each charge, and the word size of each charge. The column of features is the average size of criminal actions in the case description for each charge we use. The training set and test set in the dataset are divided into 7:3.

4.2. Experimental models

Our Model. The detailed things about our model were mentioned in Section 3.

Baselines. The main comparison model is GCI, which confirms the temporal relationship and shows a higher performance than the causal relationship between features in the judgement document. We run the same experiments as the GCI model, which runs on the 3 random seeds and reports the average (Acc) and macro-F1(F1). We split the dataset into 1%, 5%, 10%, 30% and 50%, which can study the influence of several settings of the dataset as well. Moreover, pre-training models like Bert (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and Xlnet (Yang et al., 2019) are also used as baselines. The typical deep learning model like LSTM (Graves, 2012) and Bi-LSTM (Zhou et al., 2016) are used as a baseline in the linear model.

These selected models can completely compare the superiority and stability of our model performance. Three types of models are selected: a pre-training model, an end-to-end model, and a model based on the causal relationship between features in the legal text. All use the original content of the legal text as input. The difference is that the pre-training model will re-encode and decode the legal text so that its text representation contains the semantic information of the legal text. The end-to-end model (i.e. LSTM, Bi-LSTM) uses the gate structure to analyze the legal text while the context information of the text is preserved. The innovation of GCI is that they added the Partial Ancestral Graph (PAG) that calculates the weight of the causal relationship between features and uses the Average Treatment Effect algorithm for dimensionality reduction, and then superimposes LSTM for charge prediction. CausalChain uses the causal chain in the PAG with the highest weight is selected. Both GCI and CausalChain use the logical relationship in the legal text features, so they are the main comparative models of CAG.

Ablation Study. In this part, we use all data in the dataset, and the best-performance hyper-parameter we found in experiments acts as the baseline. We designed an experiment without the SAGPool layer to support its influence of it. Furthermore, *CAG* have been divided equally into various pieces based on their locations in the CAS to affirm the principal part in the *CAG*. The epochs, the learning rate, and the weight decay are the prime hyper-parameters in the SAGPool model. These experiments discover the best scale of figures. We run the experiments with 10 random seeds and received the average Accuracy (Acc) and calculate the standard deviation (Std) in each confusing case group of experiments. The Acc indicates the performance of our method, while Std shows the stability in different case groups.

4.3. Experimental setup

The experiment in this paper is based on the Pytorch deep learning framework, and the model training is performed on the Nvidia Tesla T4 platform under the Linux system.

Bert:In this model, the pad size of the text input is set to 32, the number of Bert layers is 10, and the pre-training model used is chinese-wwm-ext-pytorch (Cui, Che, Liu, Qin, & Yang, 2021) for Chinese language processing. The hyperparameters are set to a learning rate of 2e−5 and a weight decay of 0.01.

RoBERTa:We used the roberta-base (Conneau et al., 2019) pre-training model in the RoBERTa baseline model and set the learning rate to 0.05, the maximum length of the text input vector to 128, and the other hyperparameters to their default values.

Xlnet:In the Xlnet model, the text input vector is the same as in the RoBERTa model, 128, and the learning rate is set to the model default value of 4e−5. The pre-training model, chinese-xlnet-base (Cui et al., 2020) is also used for Chinese language processing.

4.4. Evaluation metrics

For evaluating the effectiveness of prediction models, the Accuracy (Acc) and Macro-F1 are used. And we use the Standard Deviation (σ) to measure the stability of our model. Here are the interpretations of them.

In the prediction experiment, the results are split into four situations: (1)the True Positive (TP), which is predicted as true when it is really true, (2)the False Positive (FP), which is predicted as true but it is really false, (3)the False Negative (FN), which is predicted as false, but it is really true, (4)the True Negative (TN), which is predicted as false, and it is really false.

The Acc is calculated by equation (8) as follows, which means the proportion of true prediction results in all the samples:

$$\mathbf{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

The Macro-F1 is calculated by equation (9), (10) and (11) as follows, which means the reconciled mean of Precision and Recall:

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN} \tag{10}$$

$$\mathbf{F1} = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FN + FP} \tag{11}$$

The σ is calculated by equation(12), where the X_i is the number of results of each experimental epoch.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2} \tag{12}$$

5. Results

Table 2 shows the results of our model and compared models with different charges in classification. Table 3 suggests the experimental results of disambiguation performance in our model.

Table 2
Results of confusing legal charge prediction.

Models	Dataset size	F&E		AP&DD		E&MPF		Violent Acquisition		Personal Injury	
		Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
Bert	1%	76.56	38.46	53.85	35.00	–	–	64.18	40.00	79.17	29.89
	5%	82.76	48.28	78.23	74.83	–	–	69.76	22.22	77.80	–
	10%	**90.13**	87.83	**80.65**	74.83	–	–	69.57	29.91	84.04	76.19
	30%	86.73	43.76	73.24	–	–	–	74.55	43.62	83.03	29.33
	50%	91.12	73.34	71.80	75.00	–	–	76.96	72.70	82.62	58.33
RoBERTa	1%	58.82	37.04	27.28	21.42	64.29	39.13	20.58	11.38	22.72	12.46
	5%	71.08	41.54	37.73	27.39	57.97	36.70	55.29	23.73	64.55	26.15
	10%	60.60	37.73	44.85	30.96	64.02	39.03	59.11	24.76	72.73	28.07
	30%	65.98	39.73	47.81	32.34	62.89	38.06	62.19	25.56	70.09	27.47
	50%	65.99	39.75	47.19	32.06	61.41	38.04	59.25	24.81	66.39	26.60
Xlnet	1%	–	–	81.82	77.08	**92.85**	**91.81**	32.35	24.85	–	–
	5%	–	–	**83.01**	**81.32**	**89.85**	**89.53**	61.76	43.47	81.82	72.53
	10%	–	–	64.49	63.56	**89.20**	**88.33**	85.00	**81.49**	72.72	28.07
	30%	83.95	83.18	79.38	79.36	80.00	78.71	**84.51**	83.24	79.30	53.96
	50%	92.47	91.48	82.02	**82.02**	87.57	86.92	78.94	77.32	81.25	74.05
LSTM	1%	63.91	47.00	52.08	46.13	53.56	39.84	58.48	29.33	60.94	37.91
	5%	71.60	68.68	54.12	48.53	59.89	56.88	67.09	35.86	61.97	44.88
	10%	82.14	80.74	55.46	51.29	70.21	70.00	65.64	47.62	76.45	67.81
	30%	88.10	87.33	65.61	65.19	71.60	70.82	74.43	66.05	85.37	81.27
	50%	90.04	89.06	69.65	69.62	75.59	75.46	80.10	72.27	85.67	83.02
Bi-LSTM	1%	62.95	43.27	48.98	37.84	54.54	41.91	53.86	33.25	62.29	40.81
	5%	60.34	56.96	51.77	46.23	61.88	61.63	65.18	38.99	74.00	69.52
	10%	85.31	84.37	60.20	57.95	60.08	53.34	67.10	46.07	76.66	71.86
	30%	87.57	86.58	65.45	65.12	70.45	69.64	75.30	64.12	85.46	82.53
	50%	90.43	89.83	71.12	70.50	76.08	75.78	78.43	69.94	87.19	85.01
GCI	1%	82.81	82.56	62.47	61.72	74.65	70.22	57.08	42.55	67.49	49.77
	5%	88.25	87.24	78.09	77.95	83.27	83.06	69.70	60.39	81.19	75.58
	10%	**87.97**	**87.51**	78.36	78.31	85.23	84.02	74.06	67.31	80.33	74.50
	30%	**89.31**	**88.39**	80.82	80.56	88.55	88.21	75.99	70.64	84.83	80.10
	50%	90.41	89.14	81.01	80.90	89.01	88.63	76.31	71.45	85.72	81.62
CausalChain	1%	68.01	52.93	63.13	62.30	66.97	56.66	63.60	44.02	73.20	60.31
	5%	**88.64**	**87.21**	71.75	70.38	75.13	74.74	**70.57**	59.85	81.99	76.03
	10%	87.59	86.36	74.43	74.11	79.75	79.45	73.50	66.66	81.21	74.71
	30%	89.10	88.19	**80.90**	**80.50**	81.63	81.25	74.93	67.30	**85.61**	**81.00**
	50%	90.45	89.21	80.03	79.89	81.25	80.09	75.66	68.47	**86.41**	83.11
Ours	1%	**88.89**	**88.89**	**83.33**	**82.86**	76.47	75.71	**71.43**	**70.83**	**87.50**	**89.63**
	5%	75.00	74.83	77.14	75.52	89.41	89.20	70.10	**70.33**	**87.50**	**87.00**
	10%	77.78	75.92	80.00	**78.22**	88.24	88.03	**76.00**	76.00	**82.50**	**81.78**
	30%	86.11	85.87	77.64	75.52	**91.76**	**91.76**	83.67	**83.67**	80.41	80.15
	50%	**93.69**	**92.98**	**82.38**	80.83	**92.55**	**92.55**	**86.67**	**86.67**	86.04	**86.29**

Note: The bold results are the highest ones in each group, and the empty ones are overfitting. The experimental results of LSTM, Bi-LSTM, GCI and CausalChain are cited from GCI (Liu et al., 2021).

5.1. Baselines

Ours vs. Pre-processing Neural Networks. At present, natural language pre-processing neural networks such as the Bert model based on auto-encoder, the RoBERTa model, and the auto-regressive language modeling training model Xlnet that overcomes the shortcomings of the Bert model have all made good progress in text representation. It can be seen from the experimental results that both Bert and Xlnet have achieved higher experimental results in specific groups (F&E and E&MPF) when the dataset has a small amount of data. As the amount of data increases, the performance of models based on pre-processing starts to degrade. The experimental results of Bert, RoBERTa and Xlnet are about 20%, 30% and 10% lower than our model Acc and F1 respectively. In addition, on some charge sets (i.e. F&E, E&MPF), especially a small number of datasets, the pre-trained model appears to be overfitting. That is to say, the pre-training model is not as stable as the performance of CAG on a small amount of data set.

Ours vs. Typical Neural Networks. Typical neural networks, LSTM and Bi-LSTM models aim to learn the long-term dependencies in time series data. These are constantly applied to process textual data because of their well-developed construction, which prove their capabilities on marking out several types of judgement documents: however, a little insufficiently. Therefore, these two models have exposed alike performances on the scale of 1%–2% in each similar case group. Compared to our approach, we have enhanced performance by about 10% in AP&DD, E&MPF, and violent acquisition groups, and faintly higher in F&E group by around 3%.

Table 3
Disambiguation performance of our approach.

Experiments		F&E Acc±σ	AP&DD Acc±σ	E&MPF Acc±σ	Violent Acquisition Acc±σ	Personal Injury Acc±σ
Ours		**.926 ± .007**	**.751 ± .006**	**.878 ± .001**	**.843 ± .008**	**.782 ± .032**
Ours–SAGPool		.865 ± .020	.690 ± .012	.703 ± .009	.545 ± .001	.582 ± .012
Sub-CAG	Seq_{beg}	.848 ± 007	.707 ± .001	.815 ± .007	.763 ± .006	.755 ± .008
	Seq_{mid}	.817 ± .007	.693 ± .006	.807 ± .001	.593 ± .001	.730 ± .001
	Seq_{end}	.726 ± .009	.656 ± .011	.768 ± .011	.617 ± .010	.621 ± .011

Note: The first number is the average Accuracy(Acc) of 10 random seeds and the second number is their Standard Deviation(σ). The highest results are bold in each group. All the results are based on the same ratios where the epochs=200, learning rate=0.01, and weight decay=0.0001.

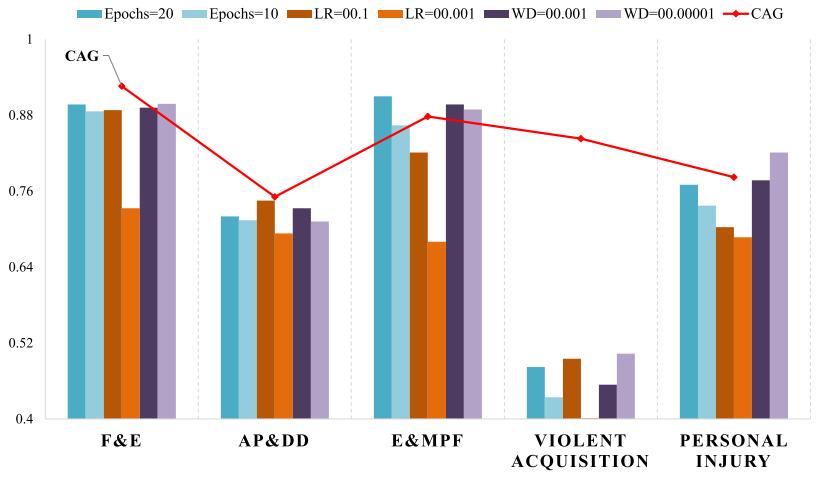


Fig. 5. This is the accuracy result of different hyper-parameter groups. The experiments are applied to illustrate that our hyper-parameter group (Epoches=200, LR=0.01, WD=0.0001) is the best chosen one.

Ours vs. GCI. It is observed that our method has raised performances in both Acc and F1 by approximately 2–3% compared with the GCI-basic model in the most group, and especially in violent acquisition group, the gap of performance has reached 10%, which indicated perfect effect in classification by using criminal actions and their temporal relationships. The CausalChain is a derivative model based on GCI, which does not use the whole text like the GCI-basic model but is applied to the casual chains that are calculated by the GCI model, which consist of wicked operations and some other crime tools or the high-frequency words within cases. The analogous achievements between GCI and CausalChain have proved that the feature exacted by casual connection could not build the characteristical representation of judge documents.

The distribution of the highest performances illustrates that, in most confusing case groups, our method has outperformed in various settings of data while in some small settings *CAG* has shown weakness, which is an acceptable fluctuation in GNNs.

5.2. The ablation study

SAGPool– layer. Our model only has 3-layers GCN when it removes the SAGPool layer, which minimizes approximately 6% in the binary classification group and drops abruptly around 20%–30% in multi-classification groups. This means the SAGPool has a capability in multi-classification within complicated textual construction, accounting for its global pooling.

The principal piece of *CAG*. The second part demonstrates that the principal information is contained at the beginning of the case description by using different parts of *CAG* in diverse locations like the beginning, midterm, or ending, which outperforms well-being among the three parts.

The supreme range of hyper-parameters. The SAPool model has three basic hyper-parameters: epochs, learning rate, and weight decay. As shown in Fig. 5, it has advanced the idea that when epochs=200, learning rate=0.01, and weight decay=0.0001, the model would expose better accuracy within *CAG* in both binary classification and multi-classification. The stability of our method has been proved in Fig. 6.

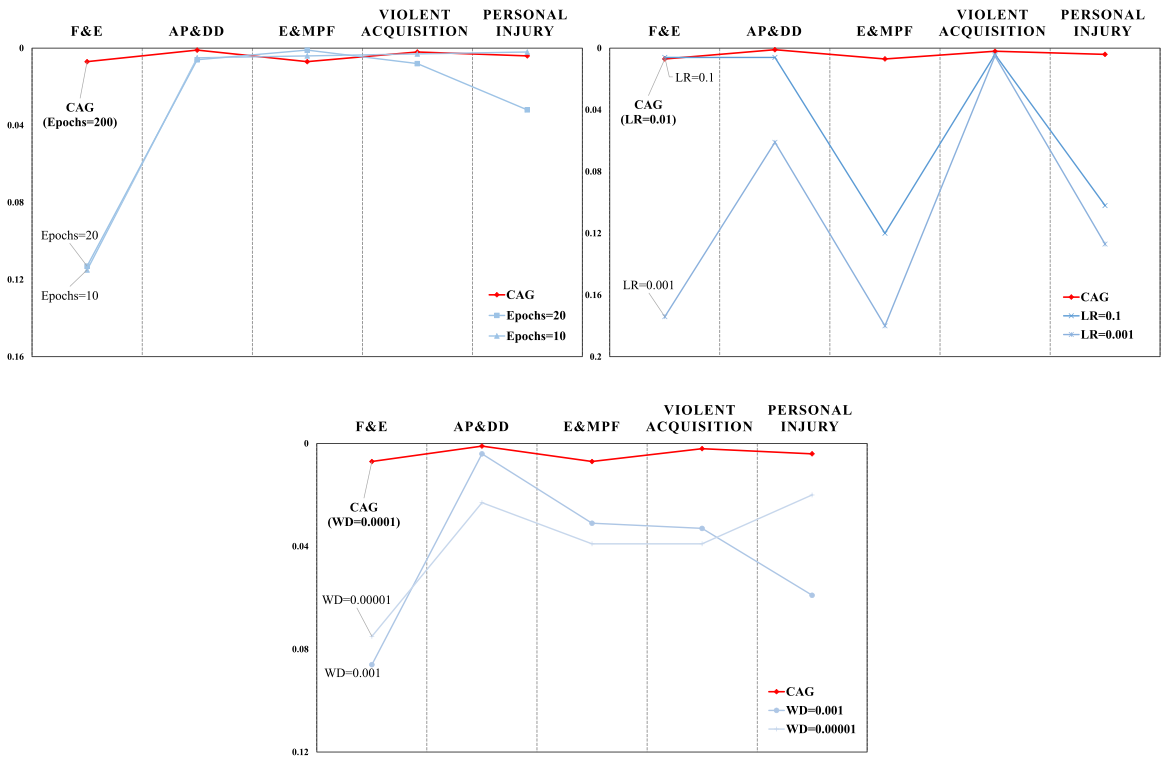


Fig. 6. This is a figure of experimental Standard Deviation with different hyper-parameter groups in each confusing cases group. The results are better when they approach 0. When the hyper-parameter group consists of Epoches=200, LR=0.01, WD=0.0001, our method performs the best stability beyond other hyper-parameter groups.

6. Analysis

6.1. The criminal action verbs in judgement documents

Unlike the related works that use features as much as possible in judgement documents, our method has merely used action verbs and their temporal relationships. According to the characteristic of judgement document, our work demonstrates that the same items shown in confusing cases, for example, money and credit cards, cannot distinguish the cases in the Fraud and Extortion case group, which will be redundant as they mislead discrimination. Even though within the *CAG* that represents textual characteristics clearly, the primary piece containing differentiated information is just a little part of it. The other explanation for their lower performances is the inappropriate input material, which contains textual overabundance while classifying the confusing cases.

Moreover, we have counted the frequency of verbs issued in the judgement document, which indicates the verb is a significant feature, owing to the criminal action that plays a central role in the judgement. Regarding the above-mentioned, we have enumerated the frequency of criminal action in the dataset, and the selected figures with actual meaning point out some constant patterns. As it exhibits in Table 3, a host of high frequency verbs have appeared simultaneously in a case group such as *in charge of* and *check* in the E&MPF group, and in the violent acquisition group, *decease* and *identify* appear in two similar cases and *damage* appears in all cases.

Relating the results in Tables 3 and 4 indicates clearly that the more different criminal action verbs the confusing cases have, the higher classified achievement they can reach. This is why the confusing cases cannot be identified exactly by making use of only the criminal action verbs, and we expand the feature representation of the judgement documents from one-dimensional sequence to *CAG* within the temporal connection of verbs to classify confusing cases accurately.

6.2. Temporal relationships in CAG

We propose that *CAG* consist of verbs about criminal actions and the temporal relationships between them to present the judge document, which can better express textual features of judgement documents than analyzing the text with causal relations between features. To prove it, we have investigated the statistical data of the dataset for each confusing case group. Criminal action verbs have been ensured, meaning, appearing simultaneously in both two or three cases in a comparable group, and counted their locations with every collected verb in the judgement document. As it shows in the heatmap, the item means time relationship between two verbs in the text, like item [*capture,transfer*] in Fig. 7 means the occurrence frequency when *transfer* comes out following *capture*.

Table 4
Statistical analysis of criminal action meaningful verbs.

Case Group	Case	Top-5 high frequency verbs of case				
F&E	Fraud	deceive	buy	provide	forge	imitate
	Extortion	catch	extort	request	exact	coerce
AP&DD	Abuse of Power	know	in charge of	demolish	check	hold a post
	Dereliction of Duty	in charge of	fell	fulfill	check	inform
E&MPF	Embezzlement	public prosecute	deceive	grant	hold a post	arbitrage
	Misappropriation of Public Funds	use for	misappropriate	public prosecute	return	hold a post
Violent Acquisition	Involuntary Manslaughter	decease	drive	pardon	damage	back a car
	Murder	decease	identify	damage	slash	stab
	Intentional Injury	damage	identify	compensate	decease	hit
Personal Injury	Robbery	identify	catch	rob	carry	steal
	Seizure	snatch	drive	rob	catch	recognize
	Kidnapping	catch	drive	hit	recognize	identify

Note: Meaningful action verbs have been chosen in the judgement document with high frequency. All the verbs are at least mentioned in half of each case group. In some case groups like F&E or E&MPF, the criminal actions are totally different. But in other case groups, only criminal action cannot distinguish the confusing cases. Therefore, it is necessary to consider the temporal relationship between criminal actions.

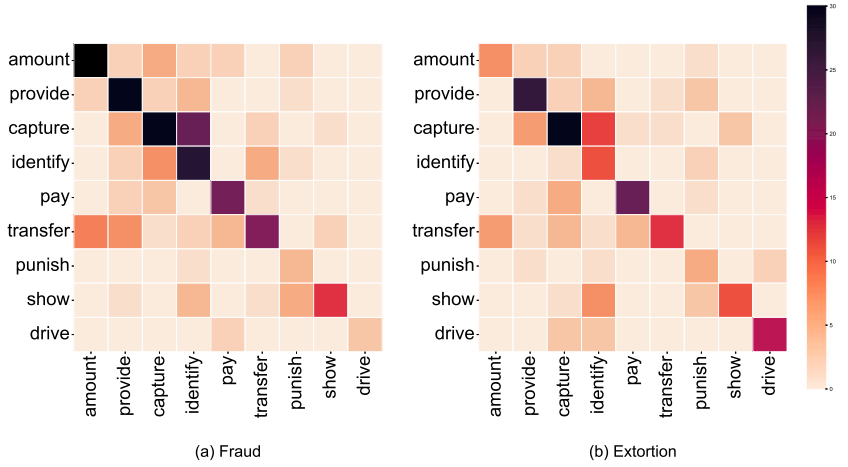


Fig. 7. This is the comparison between the temporal relationship of the F&E case group, which is shown by the heatmap. Each frame in the heatmap represents the sequence of two words, which are mentioned in all cases of the case group.

As shown in our experimental results, compared with the semantic input of CAG, the series of experiments of LSTM and bi-LSTM are unfiltered full-text input, resulting in extremely unstable performance and lack of interpretability. The semantic input model GCI, which is improved on LSTM, cannot make accurate logical judgements on case elements in limited corpus due to the complexity of its causal factors. However, the pre-trained language model has achieved better results when the amount of data is large, but for confusing cases with data skewness, especially in the case of lack of training models in the legal field, this disadvantage is even worse obvious. Therefore, it can be seen from the three series of comparative experiments that for confusing cases, the CAG model uses the temporal relationship between criminal behaviors in the text, which can analyze the judgement document more conveniently and accurately, and can also alleviate the impact of data skewness on the experimental results. The impact of the accuracy of the results, on the basis of extracting the key semantics of the judgement document, increases the difference in the text representation of confusing cases, thereby improving the accuracy of case prediction.

When we connect the heatmaps' description with the experimental result, the more different relationships when cases in the same group have, the higher classification performance they will reach, such as every group has achieved over 80% in classification accuracy in Table 2. Especially in Tables 2 and 3, F&E (Fig. 7) and E&MPF (Fig. 9) groups have reached around 90% of their extreme difference between verbs. We can find the significant verbs in different cases from heatmaps, for example, *drive* in the F&E group exhibits that all the feature verbs have a temporal relationship with it, as well as *in charge of* and *serve* in AP&DD group, etc.

6.3. Case study

We have chosen the confusing cases group in the light of obvious results, like the highest performing ones in both binary and multiple classifications, the F&E (Fig. 10.) and the Violent Acquisition group. *CAG* is beneficial for judgement document analysis. In

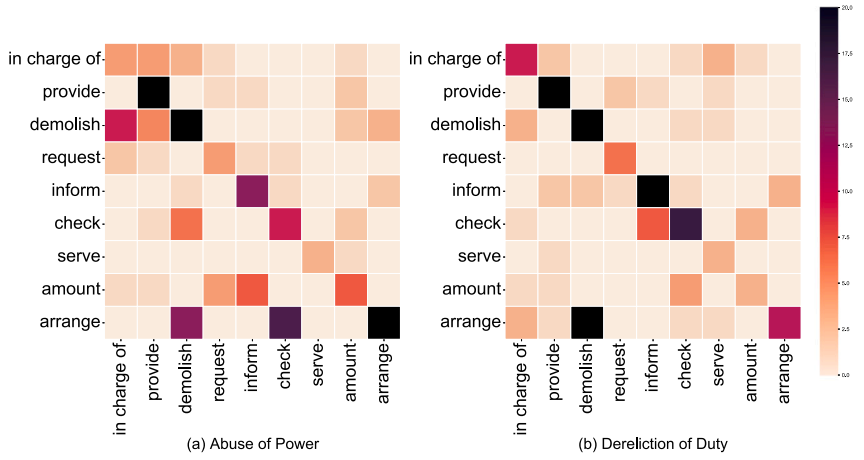


Fig. 8. These are the heatmaps of the temporal relationship between verbs in the AP&DD case group. All the settings are as same as Fig. 8.

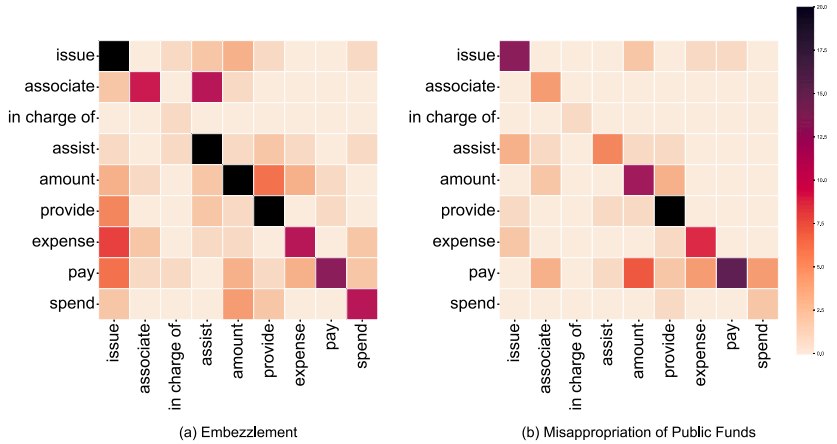


Fig. 9. These are the heatmaps of the temporal relationship between verbs in the E&MPF case group. All the settings are as same as Fig. 8.

the F&E group, even though they have a similar structure, different action verbs have supported differential semantic information for classifying confusing cases. On the contrary, when the case presentation has more similar action verbs, they can be distinguished according to their dissimilar structure, as shown in Fig. 11. In the Violent Acquisition group, seizure case and kidnapping case have alike graph structures in their *CAG*, but the differing temporal relationships between them help to predict the correct charge.

7. Discussion of results and implications

We can find that our *CAG* shows better results than GCI, which depends on the casual relationship with features in judgement document, both in terms of accuracy in criminal prediction and stability of different datasets of confusing cases, which can be made sure that the temporal relation between the criminal actions is more convincing than the causal relation calculated by the text. The main reason for the higher performance that *CAG* shows is that we catch the most important feature in the judgement document, and we save the relationship between them. Compared to concern features such as discrete variables or one-dimensional vectors, a graph that is a two-dimensional matrix can save deeper semantic information about the text. Moreover, the SAGPool model can help us filter unwanted noise of the verb node to refine the *CAG* to be more concise and make the right prediction of crime classification.

There are factors that we have not been concerned about, such as the size of the experiment dataset is not too large compared to the whole dataset and we did not distinguish different suspects in one case because we have considered that all the behaviors were acted by one suspect by default. In our dataset, most of the judgement documents are the one-crime cases, but also some multi-crime cases that we cannot classify correctly. We have already collected a few similar charges in judgement documents.

We choose the SAGPool model because of the structure of *CAG*, which have different graph topologies for each case. The SAGPool model does not consider the size and topology of graphs as input, but we have not researched whether other graph neural networks will show a higher accuracy of the classification result.

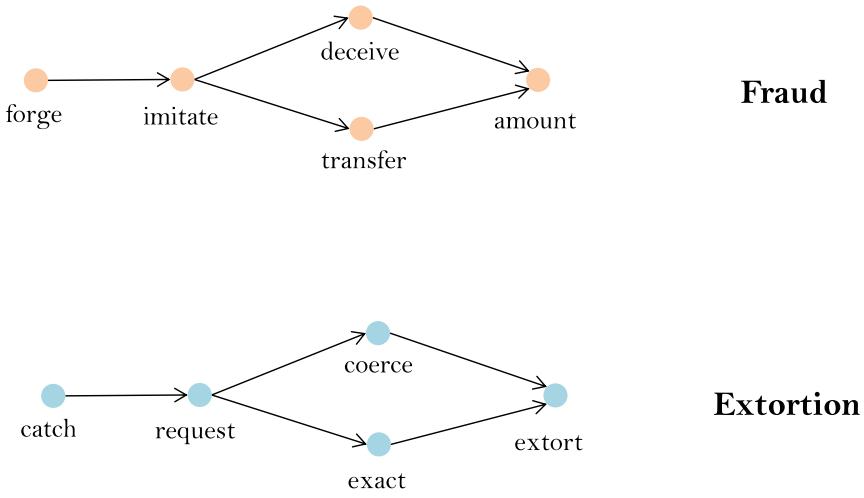


Fig. 10. This is the F&E case group's *CAG*. These two CAGs have the same structure but their node features are different, that is why they can be classified by *CAG*.

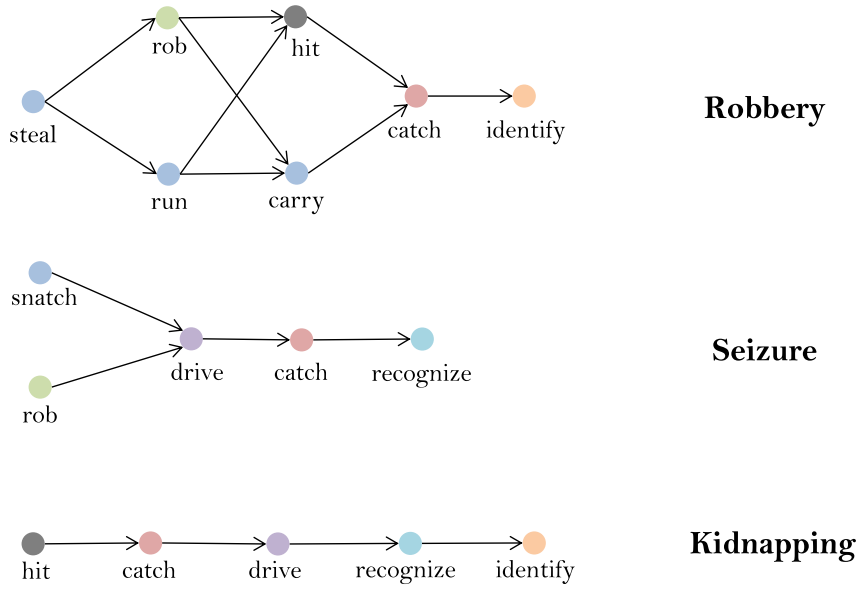


Fig. 11. This is the Violent Acquisition case group's CAG. The same actions are noted in the same action in the same color. It is clear that temporal relationships represented in *CAG* can help to distinguish confusing cases when they have similar criminal actions.

If we expand the scale of the dataset, it can summarize the mode of similar cases in different places and periods to find the regular cases, which will help not only the judges to judge cases but also help the police to prevent away from crime. Based on it, we can draw a crime mode map with past cases to predict possible crimes and avoid them.

In summary, this paper argued that the graph representation is better than the sequence when more semantic information needs to be applied. The graph expands the structure and increases the amount of information that can be processed. However, our results provide evidence for the importance of criminal actions and their temporal relationships in the judgement document. It casts a new light on mining the inner relation in legal features to assist textual analysis in the legal domain. This study provides a good starting point for discussion and further research. For the graph representation, further research is needed to confirm this novel finding, such as applying it to other legal predictions of relevant articles or terms of penalty, even the task about confusing charges, for example, similar charge search and explainable similar case matching. Regardless, future research could continue to explore more application possibilities of the *CAG*.

8. Conclusions

We propose *CAG*, a new representation of the judgement document based on criminal actions and their temporal relationships. According to the key semantic feature of the judgement documents, it contained the main information of the judgement document to improve accuracy in confusable cases classification, which has shown the value of temporal relationships between crime actions that are overlooked easily.

CRediT authorship contribution statement

Geya Feng: Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Yongbin Qin:** Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing. **Ruizhang Huang:** Formal analysis, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Yanping Chen:** Resources, Writing – review & editing, Supervision, Funding acquisition.

Data availability

Data will be made available on request

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 62066008 and No. 62066007, and the Key Technology R&D Program of Guizhou Province No. [2022] 277.

References

Aguiar, A., Silveira, R., Furtado, V., Pinheiro, V., & Neto, J. A. M. (2022). Using topic modeling in classification of Brazilian lawsuits. In *International conference on computational processing of the portuguese language* (pp. 233–242).

Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., & Sima'an, K. (2017). Graph convolutional encoders for syntax-aware neural machine translation. arXiv preprint arXiv:1704.04675.

Bi, S., Ali, Z., Wang, M., Wu, T., & Qi, G. (2022). Learning heterogeneous graph embedding for Chinese legal document similarity. *Knowledge-Based Systems*, Article 109046.

Chen, X., Jia, S., & Xiang, Y. (2020). A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, *141*, Article 112948.

Chen, H., Wu, L., Chen, J., Lu, W., & Ding, J. (2022). A comparative study of automated legal text classification using random forests and deep learning. *Information Processing & Management*, *59*(2), Article 102798.

Coelho, G. M., Ramos, A. C., de Sousa, J., Cavaliere, M., de Lima, M. J., Mangeth, A., et al. (2022). Text classification in the Brazilian legal domain. In *ICEIS (1)* (pp. 355–363).

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., et al. (2019). Unsupervised cross-lingual representation learning at scale. CoRR, abs/1911.02116. Retrieved from http://arxiv.org/abs/1911.02116.

Csányi, G., & Orosz, T. (2022). Comparison of data augmentation methods for legal document classification. *Acta Technica Jaurinensis*, *15*(1), 15–21.

Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., & Hu, G. (2020). Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: Findings* (pp. 657–668). Online: Association for Computational Linguistics, Retrieved from https://www.aclweb.org/anthology/2020.findings-emnlp.58.

Cui, Y., Che, W., Liu, T., Qin, B., & Yang, Z. (2021). Pre-training with whole word masking for Chinese BERT. *IEEE Transactions on Audio, Speech and Language Processing*, http://dx.doi.org/10.1109/TASLP.2021.3124365, Retrieved from https://ieeexplore.ieee.org/document/9599397.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Ding, X., Li, Z., Liu, T., & Liao, K. (2019). ELG: an event logic graph. arXiv preprint arXiv:1907.08015.

Erfanian, P. Y., Cami, B. R., & Hassanpour, H. (2022). An evolutionary event detection model using the matrix decomposition oriented Dirichlet process. *Expert Systems with Applications*, *189*, Article 116086.

Gao, H., & Ji, S. (2019). Graph u-nets. In *International conference on machine learning* (pp. 2083–2092).

Graves, A. (2012). Long short-term memory. *Supervised Sequence Labelling with Recurrent Neural Networks*, 37–45.

He, C., Peng, L., Le, Y., He, J., & Zhu, X. (2019). SECaps: a sequence enhanced capsule model for charge prediction. In *International conference on artificial neural networks* (pp. 227–239).

House, C. L. P. (2017). *Criminal law of the people's republic of China*.

Hu, Z., Li, X., Tu, C., Liu, Z., & Sun, M. (2018). Few-shot charge prediction with discriminative legal attributes. In *Proceedings of the 27th international conference on computational linguistics* (pp. 487–498).

Jin, W., Qu, M., Jin, X., & Ren, X. (2019). Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. arXiv preprint arXiv:1904.05530.

Jung, J., Jung, J., & Kang, U. (2021). Learning to walk across time for interpretable temporal knowledge graph completion. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 786–795).

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Lee, J., Lee, I., & Kang, J. (2019). Self-attention graph pooling. In *International conference on machine learning* (pp. 3734–3743).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Liu, X., Yin, D., Feng, Y., Wu, Y., & Zhao, D. (2021). Everything has a cause: Leveraging causal inference in legal text analysis. In *Proceedings of the 2021 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies* (pp. 1928–1941). Online: Association for Computational Linguistics.

Lyu, Y., Wang, Z., Ren, Z., Ren, P., Chen, Z., Liu, X., et al. (2022). Improving legal judgment prediction through reinforced criminal element extraction. *Information Processing & Management*, [ISSN: 0306-4573] *59*(1), Article 102780. http://dx.doi.org/10.1016/j.ipm.2021.102780, Retrieved from https://www.sciencedirect.com/science/article/pii/S0306457321002600.

Paul, S., Goyal, P., & Ghosh, S. (2022). LeSICiN: A heterogeneous graph-based approach for automatic legal statute identification from Indian legal documents. In *Proceedings of the AAAI conference on artificial intelligence, vol. 36, no. 10* (pp. 11139–11146).

Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., et al. (2003). The timebank corpus. In *Corpus linguistics, vol. 2003* (p. 40).

Qin, R., Huang, M., & Luo, Y. (2022). A comparison study of pre-trained language models for Chinese legal document classification. In *2022 5th international conference on artificial intelligence and big data* (pp. 444–449).

Rodríguez Rodríguez, C. R., Amoroso Fernández, Y., Peña Abreu, M., & Sergeevich Zuev, D. (2021). Legal decision support systems in Cuba: some background and notes for future projects. *International Review of Law, Computers & Technology*, *35*(3), 301–321.

Schneider, J. M., Rehm, G., Montiel-Ponsoda, E., Rodríguez-Doncel, V., Martín-Chozas, P., Navas-Loro, M., et al. (2022). Lynx: A knowledge-based AI service platform for content processing, enrichment and analysis for the legal domain. *Information Systems*, *106*, Article 101966.

Song, L., Wang, Z., Yu, M., Zhang, Y., Florian, R., & Gildea, D. (2018). Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks. arXiv preprint arXiv:1809.02040.

Sulis, E., Humphreys, L., Vernero, F., Amantea, I. A., Audrito, D., & Di Caro, L. (2022). Exploiting co-occurrence networks for classification of implicit inter-relationships in legal texts. *Information Systems*, *106*, Article 101821.

Sun, X., Ma, X., Ni, Z., & Bian, L. (2018). A new LSTM network model combining TextCNN. In *Neural information processing* (pp. 416–424).

Vaissnave, V., & Deepalakshmi, P. (2022). A keyword-based multi-label text categorization in the Indian legal domain using bi-LSTM. In *Soft computing: Theories and applications* (pp. 213–227).

Wenguan, W., Yunwen, C., Hua, C., Yanneng, Z., & Huiyu, Y. (2019). Judicial document intellectual processing using hy brid deep neural networks. *Journal of Tsinghua University (Science and Technology)*, *59*(7), 505–511.

Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., et al. (2018). Cail2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, *32*.

Zhang, J., Liang, S., Sheng, Y., & Shao, J. (2022). Temporal knowledge graph representation learning with local and global evolutions. *Knowledge-Based Systems*, Article 109234.

Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., & Sun, M. (2018). Legal judgment prediction via topological learning. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 3540–3549). Brussels, Belgium: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D18-1390, Retrieved from https://aclanthology.org/D18-1390.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., et al. (2016). Attention-based bidirectional long short- term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the Association for Computational Linguistics (Volume 2: Short papers)* (pp. 207–212).

Zhu, C., Chen, M., Fan, C., Cheng, G., & Zhang, Y. (2021). Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 5* (pp. 4732–4740).

ZiXian, Z. (2021). The boundary between robbery and extortion - Taking the construction of robbery as a starting point. *Law Review*, *04*, 183–196.