NeurJudge: A Circumstance-aware Neural Framework for Legal Judgment Prediction

Linan Yue[1], Qi Liu[1,*], Binbin Jin[1], Han Wu[1], Kai Zhang[1], Yanqing An[1],
Mingyue Cheng[1], Biao Yin[1], Dayong Wu[2]
[1]Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science &
School of Computer Science and Technology, University of Science and Technology of China,
{lnyue, bb0725, wuhanhan, sa517494, anyq, mycheng, ybiao912}@mail.ustc.edu.cn; {qiliuql}@ustc.edu.cn;
[2]IFLYTEK, {dywu2}@iflytek.com

ABSTRACT

Legal Judgment Prediction is a fundamental task in legal intelligence of the civil law system, which aims to automatically predict the judgment results of multiple subtasks, such as charge, law article, and term of penalty prediction. Existing studies mainly focus on the impact of the entire fact description on all subtasks. They ignore the practical judicial scenario, where judges adopt circumstances of crime (i.e., various parts of the fact) to decide judgment results. To this end, in this paper, we propose a circumstance-aware legal judgment prediction framework (i.e., NeurJudge) by exploring circumstances of crime. Specifically, NeurJudge utilizes the results of intermediate subtasks to separate the fact description into different circumstances and exploits them to make the predictions of other subtasks. In addition, considering the popularity of confusing verdicts (i.e., charges and law articles), we further extend NeurJudge to a more comprehensive framework which is denoted by NeurJudge+. Particularly, NeurJudge+ utilizes a label embedding method to incorporate the semantics of labels (i.e., charges and law articles) into facts to generate more expressive fact representations for confusing verdicts problems. Extensive experimental results on two real-world datasets clearly validate the effectiveness of our proposed frameworks.

CCS CONCEPTS

• **Applied computing** → **Law**.

KEYWORDS

Circumstances of crime; Legal judgment; Fact separation

ACM Reference Format:
Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin and Dayong Wu. 2021. 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 (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462826

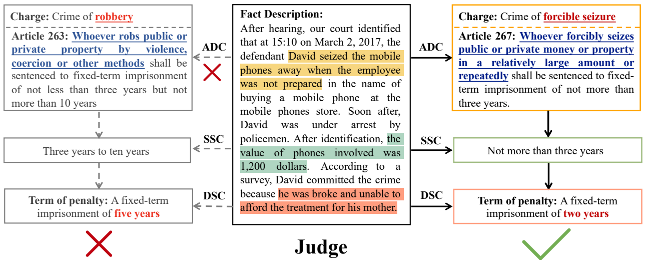

Charge: Crime of robbery
Article 263: Whoever robs public or private property by violence, coercion or other methods shall be sentenced to fixed-term imprisonment of not less than three years but not more than 10 years
ADC
SSC
DSC
Three years to ten years
Term of penalty: A fixed-term imprisonment of five years
Fact Description:
After hearing, our court identified that at 15:10 on March 2, 2017, the defendant David seized the mobile phones away when the employee was not prepared in the name of buying a mobile phone at the mobile phones store. Soon after, David was under arrest by policemen. After identification, the value of phones involved was 1,200 dollars. According to a survey, David committed the crime because he was broke and unable to afford the treatment for his mother.
Judge
ADC
SSC
DSC
Charge: Crime of forcible seizure
Article 267: Whoever forcibly seizes public or private money or property in a relatively large amount or repeatedly shall be sentenced to fixed-term imprisonment of not more than three years.
Not more than three years
Term of penalty: A fixed-term imprisonment of two years

Figure 1: An example of judgment process on charge, article, and term of penalty prediction. The right half illustrates a fine-grained judgment process. Human judges decide judgment results in sequence according to ADC, SSC, and DSC, respectively. The left half shows a misjudgment process. Since Article 263 and Article 267 are confusable, it seems to misjudge the result of articles and lead to errors in subsequent tasks easily.

1 INTRODUCTION

Based on the case descriptions, Legal Judgment Prediction (LJP) aims to predict the judgment results in the civil law system [1], including charges, law articles, and terms of penalty. It assists judiciary workers (e.g., lawyers and judges) in improving the work efficiency while ensuring the objectivity of judgment results. Besides, it also gives legal assistance to people who lack legal expertise [31, 43].

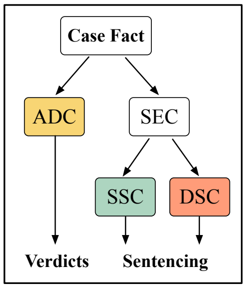
In the literature, LJP has been formalized as three text classification subtasks (i.e., charges, articles, and terms of penalty prediction) and massive efforts have been made in this area [16, 38, 43]. Among them, most methods modeled the fact description into a unified vector to predict the results of three subtasks. However, these methods still suffer from some limitations on judgment process modeling and confusing verdicts distinction.

On the one hand, when modeling the judgment process, most of the existing methods mined the impact of the entire fact description on all subtasks [37, 38, 43]. However, these methods ignore the practical judgment process of the civil law system, where human judges decide the verdicts and sentencing according to circumstances of crime [42]. Different circumstances are often located in different parts of the fact description. There exists a strict topological order in the judgment process, which follows the three fine-grained steps based on circumstances of crime. Specifically, as shown in the right

[*] Corresponding Author.

[1] The civil law system is a legal system and adopted in numerous countries (e.g., China, Germany, and France). The details of the civil law system could be found in https://en.wikipedia.org/wiki/Civil_law_(legal_system).

SIGIR '21, July 11–15, 2021, Virtual Event, Canada
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462826



(a)

Circumstances of Crime	Abbreviation	Definition
ADjudging ***C***ircumstance	ADC	A part of fact description which is in line with the definition of charge and decides the corresponding verdicts.
SEntencing ***C***ircumstance	SEC	A part of fact description which determines the terms of penalty under the premise that a certain act has constituted a crime.
Statutory ***S***entencing ***C***ircumstance	SSC	The relevant circumstances expressly provided for in the Criminal Law, which must be taken into account in sentencing.
Discretionary ***S***entencing ***C***ircumstance	DSC	Circumstances that need to be considered when sentencing according to the legislative spirit and trial practical experience.

(b)

Figure 2: (a) Relationships between Circumstances of Crime and judgment results. (b) Details about Circumstances of Crime.

half of Figure 1, human judges first decide that the defendant committed *forcible seizure* and apply *law article 267* according to the part of fact description that "*David seized the mobile phones away when the employee was not prepared*". Such part of the fact is also referred to as ***AD***judging ***C***ircumstance (ADC) which decides the corresponding verdicts (i.e., charges and law articles). Then, under the premise that the act has constituted a crime, judges determine the term of penalty according to ***SE***ntencing ***C***ircumstance (SEC) which consists of ***S***tatutory ***S***entencing ***C***ircumstance (SSC) and ***D***iscretionary ***S***entencing ***C***ircumstance (DSC). Specifically, based on the fact "*the value of phones involved was 1,200 dollars*" which is consistent with the *law article 267* and referred to as SSC, the judges decide the defendant shall be sentenced to not more than three years in prison. Finally, considering the fact "*he was broke and unable to afford the treatment for his mother.*" which refers to as DSC, the judges make the decision at their discretion (i.e., the defendant shall be sentenced to two years). More relationships and details about circumstances of crime are described in Figure 2. In summary, we could infer that the judgment process based on circumstances of crime is complex. Therefore, it is significant to design a strategy that could simulate this process and precisely represent the fact description corresponding to various circumstances.

On the other hand, there exist several confusing charges and law articles that might affect the overall performance of LJP. Specifically, due to the high similarity of charges or law articles descriptions, their corresponding verdicts could be easily misjudged, and further lead to the wrong term of penalty owing to the topological order in judgment. For example, in the left of Figure 1, the confusing *Articles 267* and *263* both describe offenses of violating property. The only difference is that the *Article 263* also describes violent behaviors while the *Article 267* not. Therefore, it is easy to misjudge the verdict as *Article 263* and significantly harm the subsequent tasks. Similarly, the problem also exists in confusing charges. Hence, it is challenging to distinguish the semantics of confusing charges and articles, and then incorporate them into the fact description to make more accurate predictions.

To tackle the above challenges, we propose a circumstance-aware neural framework (i.e., NeurJudge) for legal judgment prediction to simulate the practical judgment logic and process by utilizing circumstances of crime. According to the topological order in the judgment process, NeurJudge utilizes the prediction results of intermediate subtasks (i.e., charge and article prediction tasks) to separate the fact description into different circumstances. Specifically, it first separates the ADC and SEC from the fact according to the relevant charge, and adopts the ADC to predict the corresponding article. Next, based on the predicted article, it identifies the SSC and DSC from SEC, and utilizes them to predict the term of penalty. In addition, to alleviate the confusing verdicts problem and improve the performance of NeurJudge, we propose an extended model (denoted by NeurJudge+) on the basis of NeurJudge with a graph-based label embedding method. Particularly, we utilize the descriptions of labels (i.e., charges and articles) to construct two similarity graphs of labels. Then, we extract special label features from graphs by a decomposition strategy. With the interaction between these features and the fact description, which captures the distinguishable components in the case, more expressive fact representations are obtained and incorporated into NeurJudge to improve its performance. After that, we conduct extensive experiments on two real-world datasets to validate the effectiveness of the NeurJudge and its extension NeurJudge+ by comparing them against state-of-the-art methods. Finally, since legal AI is a sensitive field, we make some ethical discussion in the penultimate section.

2 RELATED WORK

2.1 Legal Judgment Prediction

In the early stage, many researchers focused on analyzing legal cases based on mathematical and statistical methods [12, 24, 29]. In recent years, with the development of the neural network, existing researches could be mainly divided into two lines. The first line was to utilize semantic information of law articles and charges, and the rich legal knowledge. Among them, Luo et al. [16] proposed an attention-based model by leveraging the semantics of law articles for charge prediction. Wang et al. [32] studied a pairwise attention model based on article definitions to help alleviate the label imbalance problem in law article prediction. Based on the semantics of charge, Hu et al. [9] studied the imbalanced problem and confusing charges in charge prediction by defining legal attributes of charge manually. Wang et al. [31] utilized article semantics to design a hierarchical matching network for predicting relevant articles based on the tree-shaped hierarchy where charges and articles are grouped. Zhong et al. [44] utilized legal knowledge such as elemental trial to give interpretable judgments. Zhou et al. [46] proposed a method which could project the target case to several elements on the legal knowledge graph to enhance the fact representation. Xu et al. [37] proposed a graph distillation operation to aggregate the special features from articles semantics for confusing charges problems. Besides, other researchers explored how to model the judgment process by simulating human judges. For example, Zhong et al. [43] first explored the multiple subtasks of legal judgment and modeled the judgment process according to the topological order of subtasks. Yang et al. [38] further modeled the dependencies among prediction results of multiple subtasks in LJP. Ma et al. [17] predicted the legal judgment in an encyclopedic manner by exploring the interactions between court debates and plaintiff’s claims. Although extensive research has been carried out on LJP, few considered the fine-grained judgment process where judges determined verdicts and sentencing based on different circumstances of crime.

2.2 Label Embedding

Much research existed [7, 20, 30, 40] that investigated the rich information behind class labels with label embedding methods. Among them, Wang et al. [30] embedded words and labels in the same latent space and designed an attention framework to measure the compatibility between text and labels. Chai et al. [2] proposed a framework that fed the concatenation of labels description and texts to the classifier. Besides, a few researchers explored both label semantics and structural relationships among labels. Specifically, Anthony et al. [1] utilized the description of labels and structure label spaces to solve the multi-label classification problem. Du et al. [8, 39] constructed a co-graph by exploiting the co-occurrence relationship among labels and used the Graph Convolutional Networks [11] to obtain label semantic representations for classing. The above researches have recognized the importance of the semantics of labels. However, most of these methods did not focus on the problems which have confusing semantics of labels.

3 CIRCUMSTANCE-AWARE NEURJUDGE

In this section, after formulating the LJP task as a text classification problem, we present the details of NeurJudge which is designed to simulate the practical judgment process with different circumstances of crime. Then, we describe an improved framework (i.e., NeurJudge+) for alleviating confusing verdicts problems.

3.1 Problem Definition

In this section, we show some mathematical notations and then formulate the LJP task.

Table 1 shows the corresponding notations about the input. Specifically, given the fact description s^d, the set of charge labels Y_c, article labels Y_a, and their textual definitions which offer abundant semantics, our goal is to learn a classifier ξ which is able to predict the judgment results, including charges, articles, and terms of penalty (i.e., $\{\hat{c}, \hat{a}, \hat{t}\} \Leftarrow \xi(s^d, Y_a, Y_c)$). Take the case in Figure 1 for example, based on the fact description, our ξ is to predict the charge, article and term of penalty as *Forcible Seizure*, *Article 267*, and an interval about *two to three years*, respectively.

3.2 NeurJudge Framework

In the practical judgment process, judges determine the verdicts and sentencing based on different circumstances of crime. To simulate this process, we propose a NeurJudge framework by exploiting circumstances of crime, which is shown in Figure 3(a). NeurJudge mainly consists of two components (i.e., Document Encoder and Fact Separation). Specifically, we employ document encoders to generate the semantic vectors of the text descriptions on fact, charge labels, and article labels. Then, in the Fact Separation, we propose a ***C****ircumstances of* ***C****rime aware* ***F****act* ***S****eparation (**CCFS**) method to separate fact into three parts (i.e., ADC, SSC, and DSC) based on the vectors from document encoders. Finally, we adopt them to corresponding subtasks to predict the judgment results.

Table 1: Main mathematical notations.

Notation	Description
$s^d = \left\{w_1^d, \ldots, w_{l_d}^d\right\}$	a word sequence of the fact description
$Y_c = \{c_1, \ldots, c_n\}$	the set of charge labels
$s^{c_i} = \{w_1^{c_i}, \ldots, w_{l_c}^{c_i}\}$	a word sequence of the charge c_i description
$Y_a = \{a_1, \ldots, a_m\}$	the set of article labels
$s^{a_j} = \{w_1^{a_j}, \ldots, w_{l_a}^{a_j}\}$	a word sequence of the article a_j description
$Y_t = \{t_1, \ldots, t_k\}$	the set of term of penalty labels
t_z	an arbitrary non-overlapping interval

3.2.1 *Document Encoder.*

We design the document encoders to generate the vector representations of the fact description, charge labels, and article labels. Specifically, we implement two type encoders (i.e., GRU based NeurJudge and BERT based NeurJudge).

For GRU based NeurJudge, we take the bi-directional GRU [4] as our encoder. In detail, given a word sequence of the fact description s^d, we map each word of s^d into its word embedding by adopting pre-trained word vectors, the word2vec [19], and get the word embedding sequence $\boldsymbol{E}^d = \left\{\boldsymbol{e}_1^d, \ldots \boldsymbol{e}_{l_d}^d\right\}$, $\boldsymbol{e}_i^d \in \mathbb{R}^{d_w}$, where d_w is the dimension of word embedding. For $\boldsymbol{E}^d$, we embed it into continuous hidden states by Bi-GRU encoder:

$$\boldsymbol{H}^d = \text{Bi-GRU}(\boldsymbol{E}^d), \tag{1}$$

where $\boldsymbol{H}^d = \left\{\boldsymbol{h}_1^d, \boldsymbol{h}_2^d \ldots \boldsymbol{h}_{l_d}^d\right\} \in \mathbb{R}^{l_d \times d_s}$, d_s is the double size of hidden state.

For BERT based NeurJudge, we set the BERT [5] as our encoder, and the word sequence s^d as the input. Encoded by the multi-layer self-attention structure, BERT outputs the contextual representation for each context token as $\boldsymbol{H}^d = \left\{\boldsymbol{h}_1^d, \boldsymbol{h}_2^d \ldots \boldsymbol{h}_{l_d}^d\right\} \in \mathbb{R}^{l_d \times d_s}$, where d_s denotes the dimension of the last hidden layer of BERT.

Similarly, given an arbitrary charge description s^{c_i} and article description s^{a_j}, we can obtain their hidden states $\boldsymbol{H}^{c_i} \in \mathbb{R}^{l_c \times d_s}$, $\boldsymbol{H}^{a_j} \in \mathbb{R}^{l_a \times d_s}$.

3.2.2 *Fact Separation.*

As we discussed before, judges focus on different circumstances of crime for corresponding subtasks. A visualized structure between circumstances and judgment results is shown in Figure 2(a). In the practical judgment process, judges first choose ADC from the fact to determine verdicts, and then decide the sentencing according to SEC which consists of SSC and DSC.

However, it is hard to represent the fact description corresponding to circumstances of crime which are located in various parts of the fact. From the observations of Figure 2(b), we could find that ADC is the fact which is consistent with the definition of charge. In other words, ADC is a similar component between fact description

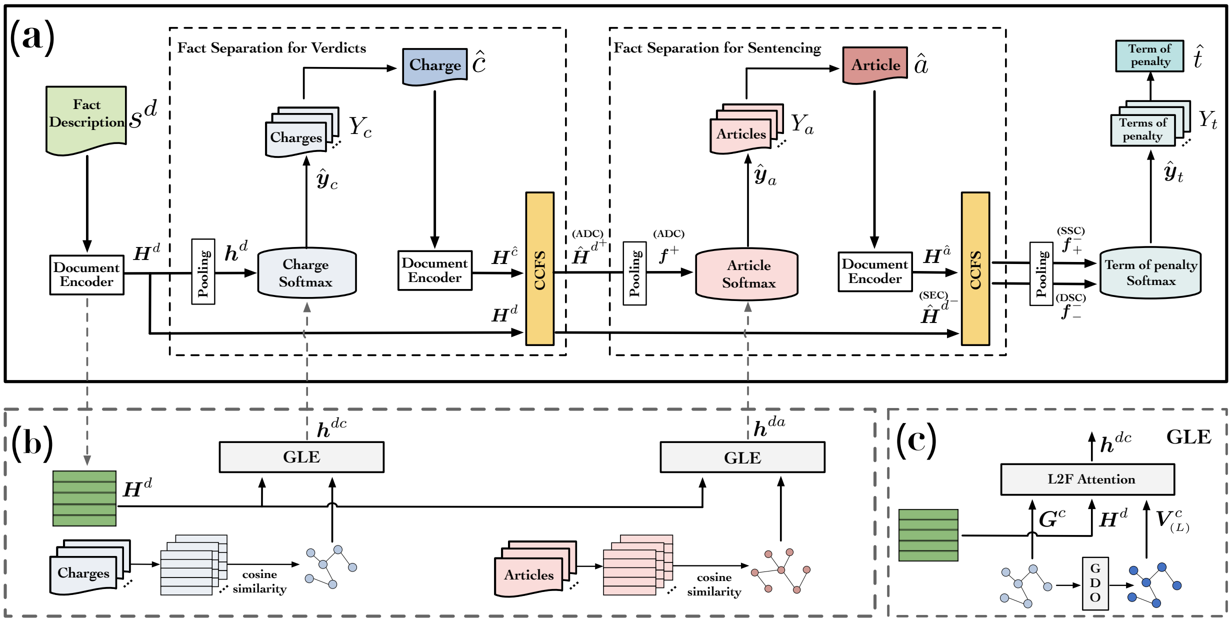


Figure 3: (a) Overview of our NeurJudge framework. (b) More expressive fact representations in NeurJudge+. (c) The Architecture of Graph-based Label Embedding (GLE) method.

and the definition of charges. Since the fact is mainly composed of ADC and SEC, SEC could be regarded as the dissimilar component. Similarly, SEC is composed of SSC and DSC, where SSC is ruled by law articles and is considered as a similar component between articles and SEC, while DSC is the dissimilar one. Inspired by these relationships between circumstances and fact, we propose a ***C****ircumstances of* ***C****rime aware* ***F****act* ***S****eparation* (**CCFS**) method to separate fact for judgment prediction.

To be specific, in fact separation for verdicts, we separate ADC and SEC aware fact representation with the definition of charge. To achieve this goal, we firstly apply per-dimension mean-pooling over $\boldsymbol{H}^d$ to obtain the final fact representation $\boldsymbol{h}^d \in \mathbb{R}^{d_s}$:

$$\boldsymbol{h}^d = \sum_{z=1}^{l_d} \boldsymbol{h}_z^d \,/\, l_d. \tag{2}$$

Next, we define an affine transformation following softmax, which is used to predict the most related charge $\hat{c}$ in the set of charge labels Y_c:

$$\hat{\boldsymbol{y}}_c = \text{softmax}\left(\boldsymbol{W}_c \boldsymbol{h}^d + \boldsymbol{b}_c\right), \tag{3}$$

where $\boldsymbol{W}_c \in \mathbb{R}^{n*d_s}$ and $\boldsymbol{b}_c \in \mathbb{R}^n$ are the trainable weight matrix and bias. Then, the most relevant charge $\hat{c}$ is computed as:

$$\hat{c} = \arg \max_{i=1,\ldots,n} \hat{y}_{c_i}, \tag{4}$$

where $\hat{y}_{c_i} \in \hat{\boldsymbol{y}}_c, i = 1, \ldots, n$. Next, we set the corresponding charge's semantic vector $\boldsymbol{H}^{\hat{c}}$ and fact vectors $\boldsymbol{H}^d$ from the *D*ocument *E*ncoder as the input of CCFS to separate case fact. Inspired by [14, 33, 34], we separate vectors into similar and dissimilar components based on *V*ector *R*ejection. Specifically, we first compute the relevance between the charge and fact that signifies which charge words are most relevant to each fact word:

$$\boldsymbol{D} = \boldsymbol{H}^d \boldsymbol{W}_f \boldsymbol{H}^{\hat{c}^\top}, \tag{5}$$

where $\boldsymbol{W}_f \in \mathbb{R}^{d_s*d_s}$. Then, we obtain $\tilde{\boldsymbol{H}}^d \in \mathbb{R}^{l_d*d_s}$ which contains the attended charges vectors for the entire fact:

$$\tilde{\boldsymbol{H}}^d = \text{softmax}(\boldsymbol{D})\, \boldsymbol{H}^{\hat{c}}. \tag{6}$$

Afterward, we apply a vector rejection operation over $\boldsymbol{H}^d$ and $\tilde{\boldsymbol{H}}^d$ to obtain the similar and dissimilar component between them:

$$\hat{\boldsymbol{H}}^{d^+} = \frac{\boldsymbol{H}^d \cdot \tilde{\boldsymbol{H}}^d}{\tilde{\boldsymbol{H}}^d \cdot \tilde{\boldsymbol{H}}^d} \tilde{\boldsymbol{H}}^d, \tag{7}$$

$$\hat{\boldsymbol{H}}^{d^-} = \boldsymbol{H}^d - \hat{\boldsymbol{H}}^{d^+}, \tag{8}$$

where $\boldsymbol{H}^d$ is separated into parallel vectors $\hat{\boldsymbol{H}}^{d^+}$ and the perpendicular ones $\hat{\boldsymbol{H}}^{d^-}$. Among them, $\hat{\boldsymbol{H}}^{d^+}$ could be seen the similar component and referred to as ADC vectors, and $\hat{\boldsymbol{H}}^{d^-}$ could be considered as the dissimilar component (i.e., SEC vectors).

Similarly, in fact separation for sentencing, we first obtain the final ADC vectors $\boldsymbol{f}^+$ by applying a mean-pooling over $\hat{\boldsymbol{H}}^{d^+}$, and define the following linear function to predict the most related article $\hat{a}$ in Y_a:

$$\hat{\boldsymbol{y}}_a = \text{softmax}\left(\boldsymbol{W}_a \boldsymbol{f}^+ + \boldsymbol{b}_a\right), \tag{9}$$

where $\hat{a} = \arg\max \hat{y}_{a_j}$, $\hat{y}_{a_j} \in \hat{\boldsymbol{y}}_a$. Next, based on $\boldsymbol{H}^{\hat{a}}$, we could separate SEC vectors $\hat{\boldsymbol{H}}^{d^-}$ into SSC vectors and DSC vectors, and apply mean-pooling operation over them to get final SSC vectors $\boldsymbol{f}_+^- \in \mathbb{R}^{d_s}$ and DSC vectors $\boldsymbol{f}_-^- \in \mathbb{R}^{d_s}$.

3.2.3 ***Prediction and Training.*** With the prediction for charges and articles (i.e., Eq. (3) and Eq. (9)), we consider the term of penalty prediction based on SSC vectors $\boldsymbol{f}_{+}^{-}$ and DSC vectors $\boldsymbol{f}_{-}^{-}$ as:

$$\hat{\boldsymbol{y}}_t = \text{softmax}\left(\boldsymbol{W}_t[\boldsymbol{f}_{+}^{-};\boldsymbol{f}_{-}^{-}] + \boldsymbol{b}_t\right), \tag{10}$$

where $\boldsymbol{W}_t$ and $\boldsymbol{b}_t$ are the parameters to learn, and " ; " represents the concatenate operation. To train this model, we use cross-entropy loss function for each subtask and take the weighted sum as an overall loss by:

$$\mathcal{L} = -\sum_{j=1}^{3}\lambda_j \sum_{k=1}^{|Y_j|} \boldsymbol{y}_{j,k}\log\left(\hat{\boldsymbol{y}}_{j,k}\right), \tag{11}$$

where $|Y_j|$ denotes the number of labels for subtask j, and λ_j is the weight factor which is the hyperparameter for each subtask.

3.3 NeurJudge+

By utilizing the prediction result of intermediate subtasks (i.e., charge and article prediction tasks), NeurJudge is modeled to deal with the problem of predicting judgment results effectively. Nevertheless, there exist several confusing verdicts (i.e., charges and articles) which affect the result of the intermediate subtasks and further limit the whole performance of NeurJudge. Therefore, to solve this problem, inspired by [15], we design an extension which could alleviate the confusing verdicts problem on the basis of NeurJudge. We denote this model by NeurJudge+ and its structure includes both Figure 3(b) and Figure 3(a) (i.e., NeurJudge). Specifically, in NeurJudge+, we firstly construct two similarity graphs of labels which are built to aggregate confusing charges and articles, respectively. Then, we propose a *G*raph-based *L*abel *E*mbedding (GLE) method consisting of *G*raph *D*ecomposition *O*peration (GDO) and *L*abel-to-*F*act (L2F) attention, which is shown in Figure 3(c). Particularly, GDO extracts label features distinguished from other similar labels on label similarity graphs. And L2F attention interacts label features with fact vectors to capture the distinguishable components of a case to enhance the final fact representation. Finally, we incorporate this representation into NeurJudge to improve the performance of charge and article prediction.

3.3.1 ***Graph Construct Layer.*** To alleviate the confusing verdicts problems, a straightforward way is to enhance the interaction between verdicts labels and case facts by label embedding methods. As the representations between similar labels are often indistinguishable, the key is to obtain the special label features by removing the similar components and retaining the dissimilar between labels. Therefore, it is significant to design a strategy to find which labels are similar. An intuitive way is to exploit the tree-shaped structure in Criminal Law of the civil law system [31]. We could consider that the children labels from the same parent label are similar. However, we find that this method ignores the relationship among the children labels from different parents. Therefore, based on the tree structure, we extend it into the graph structure to model the relationship between confusing charge and article labels.

Specifically, we set the charge similarly graph as an example, which is shown in Figure 4. There exists a hierarchical tree structure among both criminal category (denoted by **P**) and detailed charges (denoted by **c**) in Figure 4(a). On the basis of the tree structure, we first connect all children labels to construct a fully-connected graph with charge labels, as shown in Figure 4(b). Then, we calculate the weight between two nodes by cosine similarity based on their word embeddings. Finally, to focus on the most similar charges, we remove the edges with weights less than a predefined threshold τ (See Figure 4(c)). We could get the article similarly graph in the same way. To get the representations about the two graphs, based on the *D*ocument *E*ncoder which is described in the Section *NeurJudge*, we could get an arbitrary charge or article vectors $\boldsymbol{H}^{c_i}/\boldsymbol{H}^{a_j}$. After applying the per-dimension mean-pooling operation to obtain $\boldsymbol{h}^{c_i}/\boldsymbol{h}^{a_j}$, respectively, we stack all $\boldsymbol{h}^{c_i}/\boldsymbol{h}^{a_j}$ by column and obtain the original charge features $\boldsymbol{G}^c = \left\{\boldsymbol{h}^{c_1}, \ldots, \boldsymbol{h}^{c_n}\right\} \in \mathbb{R}^{n\times d_s}$, and the original article features $\boldsymbol{G}^a = \left\{\boldsymbol{h}^{a_1}, \ldots, \boldsymbol{h}^{a_m}\right\} \in \mathbb{R}^{m\times d_s}$, where c_i is the charge node and a_j is the article node.

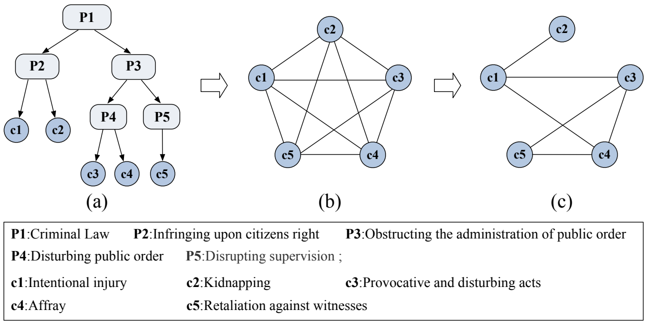

P1
P2
P3
P4
P5
c1
c2
c3
c4
c5
(a)
(b)
(c)
P1:Criminal Law **P2**:Infringing upon citizens right **P3**:Obstructing the administration of public order
P4:Disturbing public order **P5**:Disrupting supervision ;
c1:Intentional injury **c2**:Kidnapping **c3**:Provocative and disturbing acts
c4:Affray **c5**:Retaliation against witnesses

Figure 4: The charge similarity graph construction.

3.3.2 ***Graph Decomposition Operation.*** Given two graphs about $\boldsymbol{G}^c$ and $\boldsymbol{G}^a$, we adopt graph decomposition operation to extract special features of labels from these two graphs. Different from Graph Convolutional Networks [11] which may lead to the over smoothing issue [13] where the aggregated node representations would become indistinguishable especially in our similarity graphs, our GDO focuses on learning the special features of neighboring nodes. When aggregating feature information of neighboring nodes into the central node, we remove similar features between nodes and then utilize the dissimilar one to enrich the representation of the central node inspired by Xu et al. [37]. Here, we adopt the *V*ector *R*ejection operation to get similar and dissimilar features among nodes. Specifically, for an arbitrary charge node c_i in $\boldsymbol{G}^c$ at the l_{th} layer, the information aggregation from neighbors is as follows:

$$\boldsymbol{v}^{c_i}_{(l+1)} = \boldsymbol{v}^{c_i}_{(l)} - \sum_{c_j\in N_i}\frac{g(\boldsymbol{v}^{c_i}_{(l)}, \boldsymbol{v}^{c_j}_{(l)})}{|N_i|}, \tag{12}$$

$$g\left(\boldsymbol{v}^{c_i}, \boldsymbol{v}^{c_j}\right) = \frac{\boldsymbol{v}^{c_i}\cdot\boldsymbol{v}^{c_j}}{\boldsymbol{v}^{c_j}\cdot\boldsymbol{v}^{c_j}}\boldsymbol{v}^{c_j}, \tag{13}$$

where $\boldsymbol{v}^{c_i}_{(l)} \in \mathbb{R}^{d_s}$ represents the vector of c_i at the l_{th} layer. Specifically, we set $\boldsymbol{h}^{c_i}$ as $\boldsymbol{v}^{c_i}_{(1)}$ which is the representation in the first layer. N_i represents the neighbors set of c_i. The result of g is viewed as the similar component between $\boldsymbol{v}^{c_i}$ and $\boldsymbol{v}^{c_j}$. And is ta$\boldsymbol{v}^{c_i}_{(l+1)}$ken as the dissimilar component between $\boldsymbol{v}^{c_i}_{(l)}$ and its neighbors. After GDO with L layers, we output a charge node representation of the last layer (i.e., $\boldsymbol{v}^{c_i}_{(L)}$) which is the special features of c_i. Similarly, we could get an article node representation $\boldsymbol{v}^{a_i}_{(L)}$.

Table 2: The statistics of datasets.

Dataset	CAIL-small	CAIL-big
#Training Set Cases	108,619	1,593,982
#Test Set Cases	26,120	185,721
#Law Articles	99	118
#Charges	115	129
#Term of Penalty	11	11

3.3.3 *L2F Attention Layer.*

Next, inspired by [25, 41], we employ the L2F attention to alleviate confusing verdicts problems It recognizes the fact words which have the closest relation to one of the labels. Specifically, for charge labels, we first get attended fact vectors $\tilde{\boldsymbol{h}}^{dc}$ according to the original charge features $\boldsymbol{G}^c$ which aims to mitigate losses the semantics of labels, and $\hat{\boldsymbol{h}}^{dc}$ based on the special charge features $\boldsymbol{V}^c_{(L)} = \left\{\boldsymbol{v}^{c_1}_{(L)}, \ldots, \boldsymbol{v}^{c_n}_{(L)}\right\}$:

$$\tilde{\boldsymbol{h}}^{dc} = \alpha\, \boldsymbol{H}^d,\ \hat{\boldsymbol{h}}^{dc} = \beta\, \boldsymbol{H}^d, \tag{14}$$

where $\alpha \in \mathbb{R}^{l_d}$ and $\beta \in \mathbb{R}^{l_d}$ are attention vectors which are based on the original features and the special one, respectively, and they are defined as:

$$\alpha = \text{softmax}(max_{col}(\boldsymbol{H}^d \boldsymbol{W}_\alpha {\boldsymbol{G}^c}^\top)), \tag{15}$$

$$\beta = \text{softmax}(max_{col}(\boldsymbol{H}^d \boldsymbol{W}_\beta {\boldsymbol{V}^c_{(L)}}^\top)), \tag{16}$$

where $\boldsymbol{W}_\alpha \in \mathbb{R}^{d_s * d_s}$, $\boldsymbol{W}_\beta \in \mathbb{R}^{d_s * d_s}$, and max_{col} is performed across the column. Then, we concatenate $\tilde{\boldsymbol{h}}^{dc}$ and $\hat{\boldsymbol{h}}^{dc}$ to get charge label aware fact representation $\boldsymbol{h}^{dc} \in \mathbb{R}^{2d_s}$. Similarly, we can get $\boldsymbol{h}^{da} \in \mathbb{R}^{2d_s}$ for article.

3.3.4 *Prediction in NeurJudge+.*

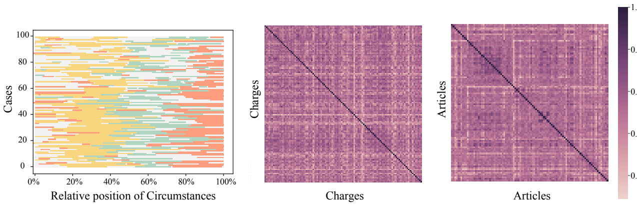
NeurJudge+ could alleviate the confusing verdicts problems by enhancing the task-specific representation of fact on charge and article prediction. Therefore, we employ the above mixed semantic vectors to predict the two subtasks. Specifically, we replace $\boldsymbol{h}^d$ with $[\boldsymbol{h}^{dc}\ ;\ \boldsymbol{h}^d]$ in Eq. (3), and replace $\boldsymbol{f}^+$ with $[\boldsymbol{h}^{da}\ ;\ \boldsymbol{f}^+]$ in Eq. (9), where " ; " represents the concatenate operation. After that, we could train NeurJudge+ by minimizing the same objective function in Eq. (11).

4 EXPERIMENTS

In this section, to demonstrate the effectiveness of NeurJudge, we first compare our NeurJudge and NeurJudge+ with some baselines and their variants on two real-world datasets. Then, we make some interpretation assessments of models.

4.1 Dataset Description

We conduct our experiments on publicly available datasets of the Chinese **AI** and **L**aw challenge (CAIL2018) which is composed of two datasets (i.e., *CAIL-small* and *CAIL-big*) [36]. CAIL2018 contains criminal cases published by the Supreme People's Court. Each case includes two parts about fact description and corresponding judgment results (i.e., charges, law articles, and terms of penalty). For data processing, referring to Zhong et al. [43], we first filter out infrequent charges and law articles and only keep those with frequencies greater than 100, and divide the terms into non-overlapping intervals. Besides, there are some cases with multiple charges and articles in real-world scenarios, which increases the complexity of judgment prediction. As our model aims to demonstrate the effectiveness of adopting circumstances of crime and be consistent with state-of-the-art methods, we filter out these multi-label samples. The detailed statistics of the datasets are shown in Table 2.



(a) The relative position of different circumstances in case facts.

(b) The similarity of charges and articles in *CAIL-small*

Figure 5: Data analysis in *CAIL-small.*

Data Analysis. We deeply analyze the *CAIL-small* dataset in Figure 5 for revealing several supportive observations of our model. Here, we first randomly sample 100 cases from *CAIL-small* and label which sentences refer to the ADC, SSC, and DSC, as there exist no corresponding labeled data. We show the relative position of various circumstances of crime in case facts in Figure 5(a). Among them, the 0% in the horizontal axis represents the beginning of cases and 100% represents the ending, and the vertical axis denotes the number of cases. The yellow, green, and pink represent the ADC, SSC, and DSC related sentences in fact description, respectively. The gray areas are elements that bore less relationship to the fact (e.g., "time" and "place"). For example, a yellow area located in 20%-45% of a certain case means the position of ADC is in the case description of between 20% and 45%. From this figure, it is evident that case facts mainly consists of circumstances of crime, and the position of circumstances in case fact is not fixed. This observation guides us that it is significant to design a strategy that could represent the fact description corresponding to various circumstances of crime. Besides, we utilize the TF-IDF [23] to extract features of charges description, and calculate the cosine similarity between charges based on these features. Figure 5(b) shows the results where the color changes from purple to pink while the value of cosines similarity decreases. From the figure, we could find that many charges are similar and confusing. These observations once again prove that similar charges could not be easily distinguished, and it is necessary to obtain special charge features. Furthermore, the result of articles similarity has the same analysis.

4.2 Baseline Methods

To evaluate the performance of our model on LJP, we adopt three representative types of baselines. First, we compare GRU based NeurJudge to some baselines on the basis of CNN or RNN, and other traditional methods:

- **Word2Vec+SVM** employs the word2vec [19] to represent word features and utilizes SVM [28] for text classification.
- **FLA** [16] is an attention-based neural network to model the interaction between fact descriptions and applicable laws.

Table 3: Judgment prediction results on CAIL-small (GRU based NeurJudge).

Methods	Charges				Law Articles				Terms of Penalty			
	Acc	MP	MR	F1	Acc	MP	MR	F1	Acc	MP	MR	F1
SVM+word2vec	83.37	80.78	77.30	78.25	84.17	80.74	75.96	77.09	33.00	25.56	25.11	22.50
FLA	84.72	83.71	73.75	75.04	85.63	83.46	73.83	74.92	35.04	33.91	27.14	24.79
TOPJUDGE	86.48	84.23	78.39	80.15	87.28	85.81	76.25	78.24	38.43	35.67	32.15	31.31
Few-Shot	88.15	87.51	80.57	81.98	88.44	86.76	77.93	79.51	39.62	37.13	30.93	31.61
LADAN	88.28	86.36	80.54	82.11	88.78	85.15	79.45	80.97	38.13	34.04	31.22	30.20
CPTP	—	—	—	—	—	—	—	—	39.16	37.06	33.82	33.79
NeurJudge	88.89	86.96	85.42	85.73	89.71	86.68	83.92	84.97	41.03	39.52	36.82	36.35
NeurJudge+	**89.92**	**87.76**	**86.75**	**86.96**	**90.37**	**87.22**	**85.82**	**86.13**	**41.65**	**40.44**	**37.20**	**37.27**

Table 4: Judgment prediction results on CAIL-big (GRU based NeurJudge).

Methods	Charges				Law Articles				Terms of Penalty			
	Acc	MP	MR	F1	Acc	MP	MR	F1	Acc	MP	MR	F1
SVM+word2vec	92.09	82.26	65.28	69.06	92.62	77.92	61.03	64.29	46.73	28.98	20.92	20.91
FLA	93.01	76.56	72.75	72.94	93.51	74.94	70.40	70.70	54.29	38.39	29.34	30.85
TOPJUDGE	93.19	79.44	75.52	75.50	93.24	74.24	71.19	70.40	53.52	44.58	30.41	30.61
Few-Shot	93.24	80.59	76.62	76.89	93.74	78.51	73.79	74.18	54.54	39.09	33.36	33.48
LADAN	93.26	81.21	77.65	77.60	93.27	75.10	72.04	71.26	53.62	41.52	37.53	36.06
CPTP	—	—	—	—	—	—	—	—	55.51	47.20	33.36	36.02
NeurJudge	95.33	84.03	77.54	78.31	95.46	81.30	75.37	76.20	55.29	44.12	35.30	36.11
NeurJudge+	**95.57**	**85.57**	**78.81**	**80.54**	**95.58**	**82.01**	**77.05**	**78.05**	**57.07**	**47.65**	**40.01**	**41.18**

- **TOPJUDGE** [43] is a topological multi-task learning model that captures the dependencies among subtasks in LJP.
- **Few-Shot** [9] is an attribute-attentive model to alleviate confusing charge issues, which utilizes the charge attributes to enhance the fact representation.
- **LADAN** [37] is an attention-based model to distinguish confusing verdicts with a graph distillation operator to learn differences between confusing law articles.
- **CPTP** [3] is a charge-based term of penalty prediction with deep gating networks which filters and aggregates charge-specified information gradually.

Then, we choose BERT and its variants which are compared with BERT based NeurJudge as follows:

- **BERT** [6] is known as a language representation built on the deep bidirectional transformers. It outperforms state-of-the-art models on a wide-range of NLP tasks. As we make experiments in Chinese datasets, we use Chinese BERT trained by [5] as our baseline method.
- **BERT-Crime** [45] is a variant of BERT, which is pre-trained with crime data.

Finally, to further validate the performance of each component in our model, we also design some simplified variants, including:

- **NeurJudge-Mtl** is a typical multi-task learning model which makes predictions for all subtasks simultaneously.
- **NeurJudge-ADC** replaces $[f_{+}^{-}; f_{-}^{-}]$ with f^{+} in Eq. (10) to predict the term of penalty, which ignores the influence of sentencing circumstance.
- **NeurJudge-GCN** replaces GDO with GCN [11] in NeurJudge+ to demonstrate the effectiveness of GDO.
- **NeurJudge-Att** expresses NeurJudge with L2F attention, which removes the GDO component of NeurJudge+.

4.3 Experimental Setup

For methods based on CNN or RNN, we first employ THULAC [27] for word segmentation as the descriptions of facts are written by Chinese with no space. Afterward, we adopt the word2vec [19] to pre-train word embeddings with embedding size 200. Meanwhile, we set the maximum document length to 350, all hidden size to 150. For methods based on BERT, we adopt the pre-trained model of Chinese which was trained by Cui et al. [5] and set the maximum document length to 500 tokens. For all methods, the weights λ_j are set as 1, and the number of spreading layers in GDO is set as 2. For training, the learning rate of the Adam optimizer [10] is initialized as 10^{-3}. We utilize PyTorch [21] to implement the proposed model and train it on a server with 2×V100 GPU, and train every model for 16 epochs with batch size 128. Finally, we employ accuracy (Acc), macro-precision (MP), macro-recall (MR), and macro-F1 (F1) as evaluation metrics to evaluate the final model[2].

4.4 Experimental Results

To demonstrate the effectiveness of our NeurJudge and NeurJudge+, we compare them with other state-of-the-art methods and their variants on charge, article, and term of penalty tasks.

[2] https://github.com/bigdata-ustc/NeurJudge

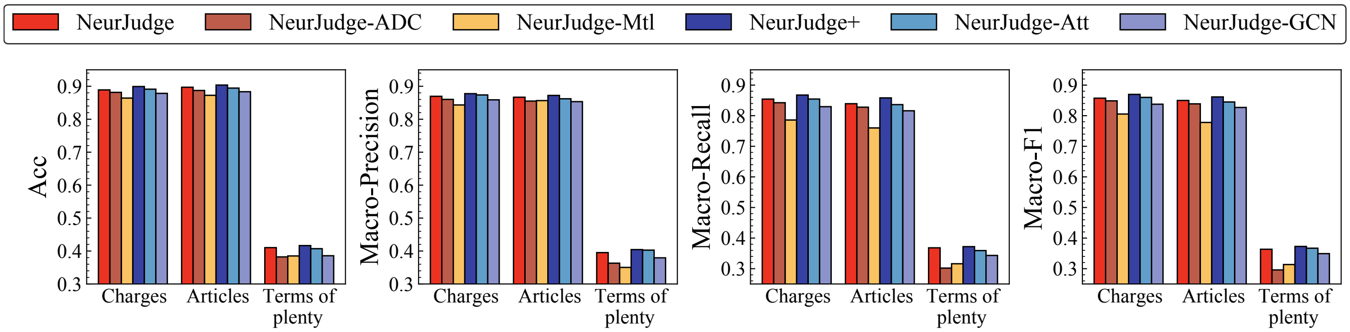


Figure 6: Results of NeurJudge and its variants of CAIL-small on four metrics (GRU based NeurJudge).

Table 5: Judgment prediction results on CAIL-small (BERT based NeurJudge).

Methods	Charges		Law Articles		Terms of Penalty	
	Acc	F1	Acc	F1	Acc	F1
BERT	90.68	87.69	90.81	86.06	40.37	34.09
BERT-Crime	91.26	87.81	91.30	85.70	40.90	34.65
NeurJudge	92.74	90.60	92.60	88.33	42.69	37.90
NeurJudge+	**92.91**	**90.89**	**92.64**	**88.75**	**43.81**	**39.76**

4.4.1 ***Comparison against baselines***. Specifically, we first compare our methods based on the GRU encoder with some baselines and the results are shown in Tables 3 and 4. We could find our proposed NeurJudge and NeurJudge+ consistently yields the best performance among all methods on two datasets. NeurJudge relatively improves over the best baseline LADAN in F1-score by 4.59% and 1.90% on average of three subtasks in *CAIL-small/big*, respectively. Furthermore, NeurJudge+ improves by 5.69% and 4.95%. To be special, from the results, we could get the following observations. (1) SVM+word2vec does not perform as well as all deep learning based models. We guess a possible reason is it fails to model the deep interactions between facts and labels. (2) FLA has poor performance because FLA is a two-stage model, which may lead the error propagation. (3) TOPJUDGE models the judgment process by utilizing the topological order of subtasks. And our NeurJudge beats it, which indicates that NeurJudge utilizes the fine-grained judgment process more effectively by focusing on different circumstances of crime for corresponding subtasks. (4) Comparing with the state-of-the-art method on the term of penalty prediction (i.e., CPTP), NeurJudge improves 2.56% and NeurJudge+ improves 3.48% on F1-score in *CAIL-small*, indicating the effectiveness of our method on simulating human judges further. (5) NeurJudge+ performs better than Few-Shot, LADAN, and NeurJudge. We believe the reason is that our proposed extension (i.e., GLE) could better extract the discriminative fact features. (6) The F1-score of all methods in *CAIL-big* dataset is worse than it in *CAIL-small* while the accuracy is better, which is because the training data of CAIL-big is highly imbalanced.

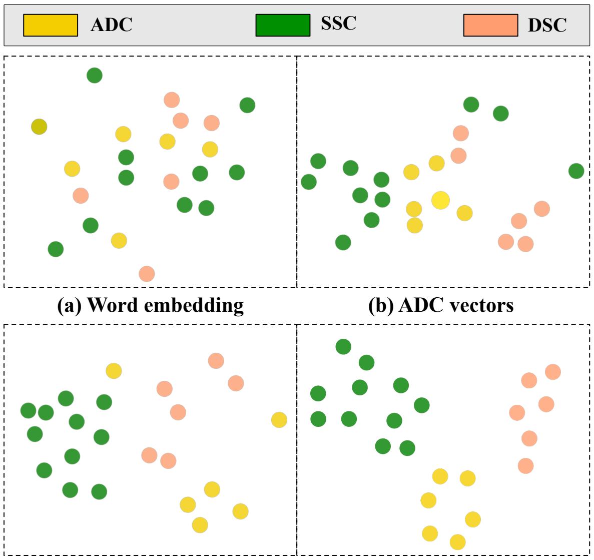
Next, we compare NeurJudge and NeurJudge+ based on the BERT encoder with BERT and its variants. As the trained cases in *CAIL-big* have reached **1,593,982** and training BERT on this dataset could be prohibitively costly, we conduct this experiment with *CAIL-small*, and the results are shown in Table 5. From the above observations, we could find BERT performs well on LJP tasks while behaves worse than our models, which further validates the effectiveness of our proposed models.

4.4.2 ***Comparison against variants of NeurJudge***. Furthermore, we analyze the impact of the CCFS and GLE components by comparing NeurJudge and NeurJudge+ with their variants on *CAIL-small*. First, we make some degeneration on NeurJudge to affirm the effectiveness of CCFS which separates the fact into different circumstances of crime to predict the corresponding subtasks. Specifically, we compare NeurJudge with NeurJudge-Mtl and NeurJudge-ADC, and relevant performances are shown in Figure 6. NeurJudge-Mtl removes the Fact Separation component, which adopts the fact representation $\boldsymbol{h}^d$ to predict all three subtasks. We could observe that NeurJudge-Mtl shows a poor performance than other methods, which demonstrates the necessity of applying different circumstances to corresponding subtasks. Comparing with NeurJudge, NeurJudge-ADC replaces $[\boldsymbol{f}^{+}_{-};\boldsymbol{f}^{-}_{-}]$ with $\boldsymbol{f}^{+}$ in Eq. (10), which ignores the influence of sentencing circumstance and only adopts ADC to predict terms of penalty. Obviously, it performs badly, especially at the term of penalty prediction, which further validates that the significance of separating circumstances of crime.

Besides, in order to verify that our proposed GLE could extract the special label features effectively, we design NeurJudge-Att and NeurJudge-GCN, and adopt them to compare with NeurJudge+ on *CAIL-small*. Specifically, we first design NeurJudge-Att method which removes the GDO and utilizes the L2F attention operation to re-encode the fact. As shown in Figure 6, we could find it performs worse than NeurJudge+ which affirms the significance of GDO. Next, we project NeurJudge-GCN method which substitutes GCN operation for GDO, and it performs the worst. It further demonstrates the GDO could effectively extract the special label features on label similarity graphs where the features of adjacent nodes are similar, and alleviate the over smoothing issue which is a serious problem in GCN.

4.5 Case Study

We provide a qualitative analysis of NeurJudge and NeurJudge+. First, NeurJudge adopts the CCFS method to separate the fact vector into three parts (i.e., ADC vectors, SSC vectors, and DSC vectors). In order to intuitive see the difference among the above vectors, we visualize these features spaces using t-SNE [18] in Figure 7.



(a) Word embedding (b) ADC vectors

(c) SSC vectors (d) DSC vectors

Figure 7: T-SNE visualizations of fact vectors.

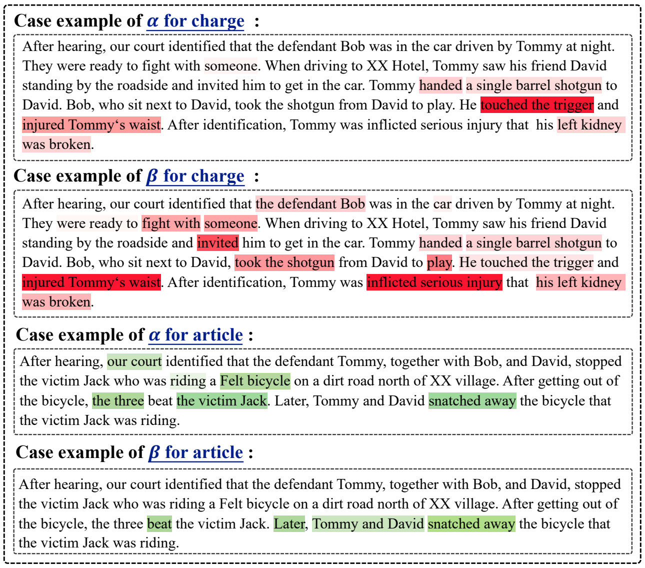
Specifically, we visualize the case which is illustrated in Figure 1. The yellow points in Figure 7 represent the circumstances related words which are highlighted with yellow in Figure 1, similar to the green and the pink. Figure 7(a) shows the word embedding space of these circumstances, as we can see, words in it are highly fragmented. After the fact separation for verdicts, we could find the words about ADC (the yellow) are gathered together in Figure 7(b). Next, we separate sentencing circumstances further. From the observations of Figure 7(c-d), SSC and DSC vectors can be separated and gathered obviously. The above visualizations demonstrate that NeurJudge could well separate different circumstances of facts.

Next, to intuitively verify that the extension (i.e., GLE) extracts special features effectively, we visualize attention vectors α and β (i.e., Eq. (15-16)) in Figure 8, where the darker a word is, the higher the attention weight it gets. We choose two case examples for charge and article, respectively, each of them has α and β visualizations. Specifically, for the first case, its actual charge label is ***crime of negligence causing serious injury***. However, the α vector only focuses on words like *"shotgun"* and *"injured"*, while the β assigns heavier weights to *"play"* additionally which shows the shooting is negligent. Similarly, in the second example, its actual article label is ***Article 263*** which describes the violation of the health and property rights. As we can see, the α vector does not focus on the word *"beat"* which shows the violent acts, but the β does. From the above observations, our GLE could well extract the special label features and capture the distinguishable components in the fact for charge and article prediction.

5 ETHICAL DISCUSSION

Since the results of legal judgment involves the practicality of the litigant, and LegalAI is an emerging but sensitive technology, there are certain ethical concerns worth discussing.

Although NeurJudge and NeurJudge+ have achieved excellent performance on real datasets, it is still worth noting that the method or the system is not intended to replace offline criminal litigation, nor is it to replace the independent judgment of judicial personnel. It is assistance for human judges, which can help judges adopt the relevant law articles quickly and play a huge role in ensuring the principle of *"treating like cases alike"* [22, 26].



Case example of *α* for charge :

After hearing, our court identified that the defendant Bob was in the car driven by Tommy at night. They were ready to fight with someone. When driving to XX Hotel, Tommy saw his friend David standing by the roadside and invited him to get in the car. Tommy handed a single barrel shotgun to David. Bob, who sit next to David, took the shotgun from David to play. He touched the trigger and injured Tommy's waist. After identification, Tommy was inflicted serious injury that his left kidney was broken.

Case example of *β* for charge :

After hearing, our court identified that the defendant Bob was in the car driven by Tommy at night. They were ready to fight with someone. When driving to XX Hotel, Tommy saw his friend David standing by the roadside and invited him to get in the car. Tommy handed a single barrel shotgun to David. Bob, who sit next to David, took the shotgun from David to play. He touched the trigger and injured Tommy's waist. After identification, Tommy was inflicted serious injury that his left kidney was broken.

Case example of *α* for article :

After hearing, our court identified that the defendant Tommy, together with Bob, and David, stopped the victim Jack who was riding a Felt bicycle on a dirt road north of XX village. After getting out of the bicycle, the three beat the victim Jack. Later, Tommy and David snatched away the bicycle that the victim Jack was riding.

Case example of *β* for article :

After hearing, our court identified that the defendant Tommy, together with Bob, and David, stopped the victim Jack who was riding a Felt bicycle on a dirt road north of XX village. After getting out of the bicycle, the three beat the victim Jack. Later, Tommy and David snatched away the bicycle that the victim Jack was riding.

Figure 8: Attention visualizations of GLE.

As mentioned before, the judgment prediction is an emerging technology and there exist some risks at its exploratory stage. The goal of our algorithm is to give the charge, article and term of penalty of cases, but whether the algorithm makes appropriate analysis of cases remains doubtful. Judges need to check the judgment results from algorithms [35].

6 CONCLUSIONS

In this paper, we proposed a circumstance-aware framework (i.e., NeurJudge) which simulated the judgment process to improve the performance of LJP. To be specific, by utilizing the results of intermediate subtasks, NeurJudge employed CCFS to separate the fact into different circumstances and utilized the representations of these circumstances to predict corresponding tasks. In addition, to address confusing verdicts issues, we designed an extension on the basis of NeurJudge denoted by NeurJudge+. Particularly, with the interaction between the extracted label features from the label similarity graphs and the fact description, more expressive fact representations have been incorporated into NeurJudge to make more accurate predictions. Extensive experiments on two real-world datasets demonstrated the superiority of NeurJudge and NeurJudge+. Finally, we made ethical discussions of our work since the sensitivity and particularity of Legal AI.

7 ACKNOWLEDGEMENTS

This research was partially supported by grants from the National Key Research and Development Program of China (No. 2018YFC0832101), and the National Natural Science Foundation of China (Grants No. 61922073 and 71802068).

REFERENCES

[1] Anthony, Rios, Ramakanth, and Kavuluru. 2019. Few-Shot and Zero-Shot Multi-Label Learning for Structured Label Spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[2] Duo Chai, Wei Wu, Qinghong Han, Fei Wu, and Jiwei Li. 2020. Description based text classification with reinforcement learning. In *Proceedings of 37th International Conference on MachineLearning (ICML)*.

[3] Huajie Chen, Deng Cai, Wei Dai, Zehui Dai, and Yadong Ding. 2019. Charge-based prison term prediction with deep gating network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 6363–6368.

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

[5] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-Training with Whole Word Masking for Chinese BERT. *arXiv preprint arXiv:1906.08101* (2019).

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. [n.d.]. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 4171–4186.

[7] Cunxiao Du, Zhaozheng Chen, Fuli Feng, Lei Zhu, Tian Gan, and Liqiang Nie. 2019. Explicit interaction model towards text classification. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Vol. 33. 6359–6366.

[8] Yichao Du, Pengfei Luo, Xudong Hong, Tong Xu, Zhe Zhang, Chao Ren, Yi Zheng, and Enhong Chen. 2021. Inheritance-guided Hierarchical Assignment forClinical Automatic Diagnosis.. In *arXiv preprint arXiv:2101.11374*.

[9] Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Few-shot charge prediction with discriminative legal attributes. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*. 487–498.

[10] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

[11] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

[12] Fred Kort. 1957. Predicting Supreme Court decisions mathematically: A quantitative analysis of the" right to counsel" cases. In *The American Political Science Review*, Vol. 51. JSTOR, 1–12.

[13] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the 32st AAAI Conference on Artificial Intelligence*.

[14] Qi Liu, Zai Huang, Zhenya Huang, Chuanren Liu, Enhong Chen, Yu Su, and Guoping Hu. 2018. Finding similar exercises in online education systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1821–1830.

[15] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2019. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering* 33, 1 (2019), 100–115.

[16] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to Predict Charges for Criminal Cases with Legal Basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2727–2736.

[17] Luyao Ma, Wei Ye, and Shikun Zhang. 2020. Judgment Prediction Based on Case Life Cycle. In *The 1st International Workshop on Legal Intelligence Held in conjunction with SIGIR 2020 (LegalAI@SIGIR2020)*.

[18] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. In *Journal of machine learning research*, Vol. 9. 2579–2605.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[20] Nikolaos Pappas and James Henderson. 2019. Gile: A generalized input-label embedding for text classification. In *Transactions of the Association for Computational Linguistics*, Vol. 7. MIT Press, 139–155.

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*. 8026–8037.

[22] Christopher Rigano. 2019. Using artificial intelligence to address criminal justice needs. *National Institute of Justice Journal* 280 (2019), 1–10.

[23] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[24] Jeffrey A Segal. 1984. Predicting Supreme Court cases probabilistically: The search and seizure cases, 1962-1981. In *American Political Science Review*, Vol. 78. Cambridge University Press, 891–900.

[25] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

[26] Changlong Sun, Yating Zhang, Q. Zhang, and Xiaozhong Liu. 2020. Legal Artificial Intelligence - Have You Lost a Piece from Jigsaw Puzzle?. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*.

[27] Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu. 2016. Thulac: An efficient lexical analyzer for chinese.

[28] Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. In *Neural processing letters*, Vol. 9. Springer, 293–300.

[29] S Sidney Ulmer. 1963. Quantitative analysis of judicial processes: Some practical and theoretical applications. In *Law and Contemporary Problems*, Vol. 28. JSTOR, 164–184.

[30] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint Embedding of Words and Labels for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2321–2331.

[31] Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. 2019. Hierarchical matching network for crime classification. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 325–334.

[32] Pengfei Wang, Ze Yang, Shuzi Niu, Yongfeng Zhang, Lei Zhang, and ShaoZhang Niu. 2018. Modeling dynamic pairwise attention for crime classification over legal articles. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 485–494.

[33] Xin Wang, Wei Huang, Qi Liu, Yu Yin, Zhenya Huang, Le Wu, Jianhui Ma, and Xue Wang. 2020. Fine-Grained Similarity Measurement between Educational Videos and Exercises. In *Proceedings of the 28th ACM International Conference on Multimedia*. 331–339.

[34] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence Similarity Learning by Lexical Decomposition and Composition. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*. 1340–1349.

[35] Yiquan Wu, Kun Kuang, Yating Zhang, Xiaozhong Liu, Changlong Sun, Jun Xiao, Yueting Zhuang, Luo Si, and Fei Wu. 2020. De-biased Court's View Generation with Causality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 763–780.

[36] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. In *arXiv preprint arXiv:1807.02478*.

[37] Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish Confusing Law Articles for Legal Judgment Prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

[38] Wenmian Yang, Weijia Jia, XIaojie Zhou, and Yutao Luo. 2019. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*.

[39] DU Yichao, XU Tong, MA Jianhui, CHEN Enhong, Tongzhu LIU Yi ZHENG, and TONG Guixian. [n.d.]. An automatic ICD coding method for clinical records based on deep neural network. *Big Data Research* 6, 5.

[40] Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2017. Multi-task label embedding for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[41] Kai Zhang, Hao Qian, Qing Cui, Qi Liu, Longfei Li, Jun Zhou, Jianhui Ma, and Enhong Chen. 2021. Multi-Interactive Attention Network for Fine-grained Feature Learning in CTR Prediction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 984–992.

[42] Mingkai Zhang et al. 2003. *Criminal Law*. Number 4. Law Press·China. 121–128,502–513 pages.

[43] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal Judgment Prediction via Topological Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[44] Haoxi Zhong, Yuzhong Wang, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Iteratively questioning and answering for interpretable legal judgment prediction. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, Vol. 34. 1250–1257.

[45] Haoxi Zhong, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2019. *Open Chinese Language Pre-trained Model Zoo*. Technical Report. https://github.com/thunlp/openclap

[46] Xin Zhou, Yating Zhang, Xiaozhong Liu, Changlong Sun, and Luo Si. 2019. Legal Intelligence for E-commerce: Multi-task Learning by Leveraging Multiview Dispute Representation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 315–324.