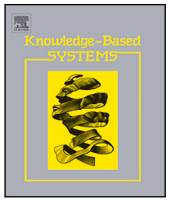
Knowledge-Based Systems 299 (2024) 112033

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



HD-LJP: A Hierarchical Dependency-based Legal Judgment Prediction Framework for Multi-task Learning

Yunong Zhang [a], Xiao Wei [a,b,*], Hang Yu [a,b]

[a] School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China
[b] Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, Shanghai University, Shanghai, 200444, China

ARTICLE INFO

Keywords:
Legal judgment prediction
Task judicial logic
Label topological relation
Hierarchical semantics knowledge
Multi-task learning

ABSTRACT

In real-world scenarios, multiple subtasks of legal judgment (such as law article, charge, and term of penalty) have strict task logical order and label topological relation, and their results influence and validate each other. However, most existing methods model them as simple classification problems, which ignores the logical and semantic constraints between subtasks. Besides, they mainly focus on the fact description for judgment results, and ignore the standard legal documents (i.e., the established law articles). To this end, we propose a **H**ierarchical **D**ependency-based **L**egal **J**udgment **P**rediction framework (HD-LJP), which integrates task judicial logic, label topological relation, and hierarchical semantics knowledge in legal text effectively. Specifically, HD-LJP employs consistency and distinction distillation to model label topological relation among multiple subtasks, and improve the differentiation of each subtask itself respectively. In addition, for simulating the judicial logic of human judges, we define logical dependencies between subtasks, and then utilize the results of intermediate subtasks to make auxiliary prediction of other subtasks. And, hierarchical semantics knowledge is fully integrated and applied in these two processes, which will profoundly affect the creditability and interpretability of the judgment results. The experimental results show that HD-LJP can improve the prediction performance of three LJP subtasks, especially in the term of penalty. Compared with the existing methods, the macro-F1 on CAIL-small is increased by 13.4%, and 6.9% on CAIL-big. In addition, through further case studies, this paper demonstrates that HD-LJP performs better for tail classes and confusing labels than the current SOTA R-former.

1. Introduction

In recent years, artificial intelligence technology has been extensively applied in legal text mining [1,2], which can improve the work efficiency, enhance the case understanding ability, and can carry out the auxiliary decision tasks [3]. Among them, legal judgment prediction (LJP) mainly concerns how to make correct judgment predictions (e.g., law articles, charges, and terms of penalty.) based on the case descriptions [4]. It is essential for the legal assistance system, not only to provide legal consulting services for the general public but also to provide objective judgment references for professionals (e.g., lawyers and judges). Therefore, how to make more accurate and practical legal judgment prediction by utilizing artificial intelligence technologies has drawn lots of attention in intelligent judicial research.

As illustrated in Fig. 1, in real-world scenarios, these multiple subtasks in LJP have strict logical order and topological relation for human judges, and their results influence and validate each other [5]. Specifically, given the fact description of a specific case, charge description, and established law articles, the judge first identifies the relevant law articles based on the case description, and subsequently determines the charges by referring to the statutory articles [6]. Based on the judgment results mentioned above, the judge can further determine the sentence. Thus, legal judgment prediction is essentially a multi-task learning (MTL) problem.

In summary, how to simulate the task judicial logic and model the topological relation of subtask labels will deeply influence the creditability of judgment results. However, most of the existing methods still struggle with three major challenges, prompting us to develop a novel prediction model based on the fact description, charge description, established law articles, task judicial logic, and label topological relation. Specifically, it includes the following three challenges:

Integrating legal text semantic knowledge. The majority of existing studies [5–7] focus on the factual description, and in fact, human judges also adopt standard legal documents (i.e., established law articles and charge descriptions) to decide judgment results. Especially in penalty prediction, the content of legal provisions has stipulated the scope

* Corresponding author at: School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China.
E-mail addresses: ynzhang0168@shu.edu.cn (Y. Zhang), xwei@shu.edu.cn (X. Wei), yuhang@shu.edu.cn (H. Yu).

https://doi.org/10.1016/j.knosys.2024.112033
Received 4 December 2023; Received in revised form 10 May 2024; Accepted 29 May 2024
Available online 31 May 2024

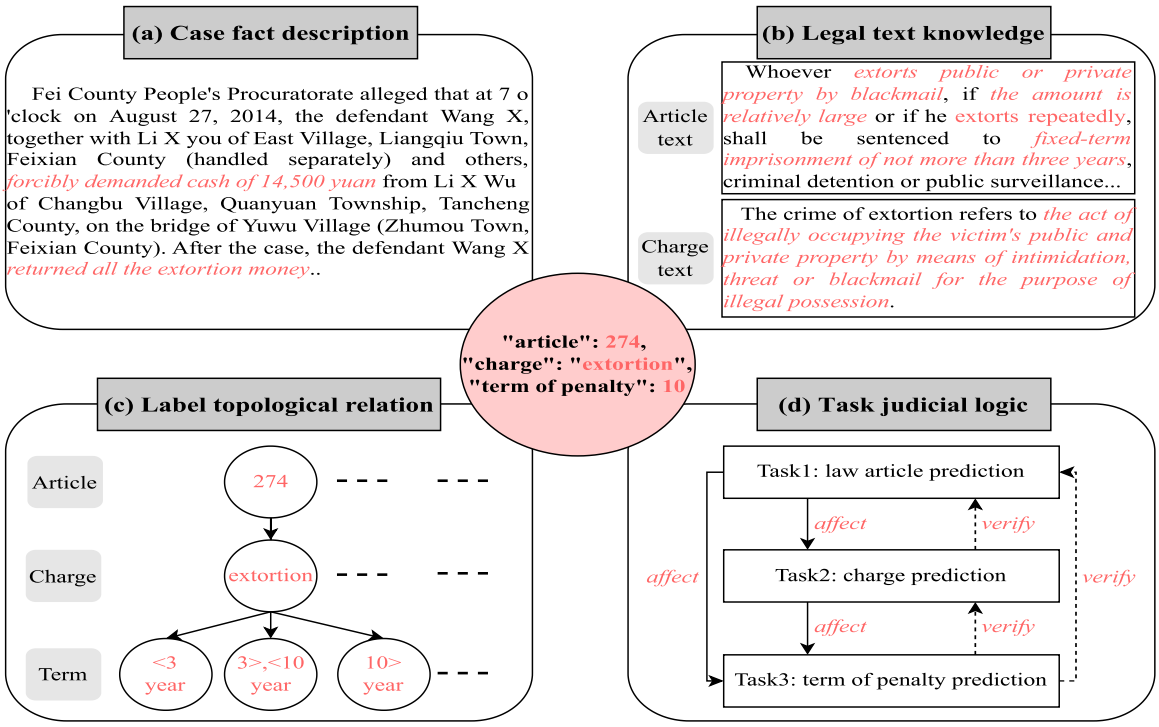


Fig. 1. A Case of Legal Judgment Prediction with four auxiliary information.

of the applicable period of a crime [8,9]. For instance, as shown in Fig. 1(b), whoever extorts public or private property by blackmail, if the amount is relatively large or if he extorts repeatedly, shall be sentenced to fixed-term imprisonment (less than three years), criminal detention, or public surveillance. Thus, incorporating the semantic knowledge from law articles and charge descriptions will improve the performance of the sentence prediction.

Modeling label topological relation. In established law articles, the label relationship of different judgment tasks exists naturally. For example in Fig. 1(c), the charge "*Crime of Extortion*" is only related to the law article *274*. And, the term mentioned in charge "*Crime of Extortion*" is fixed-term imprisonment (less than three years), if the amount of extortion is relatively large or repeated extortion. However, existing methods [10,11] usually ignore the hierarchical structure of Multiple judgment subtasks, and feed them into a classification framework to make predictions. They are not guaranteed to meet these constraints, which may lead to conflicting judgment results. Certainly, it is critical to fuse hierarchical structure and semantic information of labels into a unified framework to model topological relation for the correct verdict [12].

Simulating task judicial logic. A strict logical order exists among the subtasks of legal judgment [5,7] in the Civil Law System, and their prediction results influence and validate each other. As illustrated in Fig. 1(d), human judges first determine the applicable law article based on case facts, and then determine the charge according to the law article and case facts. Based on the above results, the judge further determines the length of the sentence. However, most multi-task learning methods generally focus on sharing representations or some encoding layers among relevant tasks [8,11,13–15], they ignore the interaction and dependencies among multi-task results [6,16,17]. Therefore, it is meaningful to exploit the result of the intermediate task to extrapolate other tasks based on the judgment logical order.

To address these challenges, we introduce HD-LJP to simulate the practical judgment process, which integrates fact description, task judicial logic order, label topological relation, and hierarchical semantics knowledge in legal text effectively. First of all, HD-LJP provides a unified representation of the hierarchical semantics knowledge contained in the established law articles, which includes law articles, charges, and the terms of penalty. Secondly, we initialize the label semantics of multiple tasks employing the fusion representation of fact description and the above hierarchical semantic knowledge. Further, HD-LJP employs consistency and distinction relation distillation to model label topological relation among multiple subtasks and improve the differentiation of each subtask itself respectively. In addition, for simulating the task judicial logic of human judges, we utilize the results of intermediate subtasks to make auxiliary predictions of other subtasks by adopting a novel attention mechanism based on case facts. Finally, on two CAIL-2018 datasets [18], we verify the effectiveness and robustness of the proposed method.

To sum up, this work has four key contributions, outlined as follows:

- Based on the task judicial logic, label topological relation, and hierarchical semantics knowledge in legal text, we propose a novel hierarchical dependency-based LJP framework, named HD-LJP, to jointly predict multiple subtasks.
- To exploit effective label semantic knowledge from existing law articles and charge description, we employ consistency and distinction distillation to model the label topological relation among multiple subtasks, and improve the differentiation of each subtask itself respectively.
- To simulate the judicial logic of human judges, we define the dependencies between LJP subtasks, and then utilize the results of intermediate subtasks to make auxiliary predictions of other subtasks by adopting a attention mechanism based on case facts.
- The experiments on two real CAIL-2018 datasets show that our HD-LJP achieves the best performance compared to all the baselines on three subtasks, especially on the penalty prediction, where HD-LJP achieves a 13.34% relative improvement on CAIL-small and 6.9% on CAIL-big for macro-F1.

The rest of this paper is organized as follows. After a summary of related works in Section 2, we give the problem formalization and the motivation of our work in Section 3. In Section 4, we describe our proposed framework and four important component modules. Then, we

provide experimental results and evaluation analysis in Section 5. Finally, Section 6 summarizes this work and explains the future research direction.

2. Related works

In this section, we survey relevant works of legal judgment prediction, which mainly include feature knowledge learning and multi-task relation learning.

2.1. Feature knowledge learning

Earlier works often focused on legal cases in specific scenarios with machine learning algorithms and defined LJP as a text classification problem [19–21]. However, these methods are limited to shallow textual features and manually designed factors, which is insufficient for fully extracting semantic information from case facts at multiple levels. In recent years, with the development of artificial intelligence technology, more and more researchers are trying to integrate deep neural networks with legal knowledge [22–24], such as case features, article descriptions, charge descriptions, etc.

For studying the imbalanced problem and predicting confusing charges, Hu et al. [25] manually defined ten discriminative attributes to enhance the representation of the fact description. Similarly, Guo et al. [26] extracted nine auxiliary sentences via judicial domain knowledge to help predict low-frequency charges. Xu et al. [27] also suggested integrating case keywords into the LJP framework for multi-task learning. In addition, using the first-order logic rules, Gan et al. [28] integrated legal knowledge into a co-attention based network for judgment prediction.

Nevertheless, these methods are designed for specific subtasks and cannot be easily extended to more subtasks of LJP. Therefore, to jointly model multiple subtasks, Yang et al. [8] proposed a multi-task prediction framework based on the multi-view encoder fusing legal keywords. Xu et al. [13] extracted differences between legal provisions to enhance the representation of case facts for confusing charges problems. Luo et al. [10] presented an attention hierarchical network, which jointly models fact descriptions and relevant law articles to improve the charge performance. Wang et al. [29] studied a unified dynamic pairwise model based on law article definitions, which can help alleviate the label imbalance problem.

Furthermore, to take full advantage of the label semantics and the label structure, Wang et al. [12] designed a hierarchical matching network to predict relevant law articles. However, they do not make full use of legal information, and most of them are only for single-label problems. Zhong et al. [30] proposed a law article element-aware model, which manually divides and labels each article into seven elements. Besides, based on exploring charge labels, Ye et al. [31] proposed a label-conditioned Seq2Seq model to generate court views, and predict charges in Chinese civil law. Chen et al. [32] proposed a deep gating network for a charge information filter, which significantly improves the accuracy of charge-based term prediction.

However, they ignore that judges apply various parts of case facts to determine the outcome. To this end, by exploring circumstances of crime, Yue et al. [17] introduced a circumstance-aware LJP framework, and incorporated the label semantics of law articles and charges to generate more expressive fact representations. Following the procedures of a human judge, Long et al. [33] applied reading comprehension framework to capture the complex semantic interactions among fact description, plaintiffs' claim and law articles.

Although the above methods have been proven to be effective, few considered the fine-grained label or task dependency of legal auxiliary knowledge, which limits their transferability in multiple subtasks of legal judgment prediction.

2.2. Multi-task relation learning

Usually, legal judgment mainly consists of law articles, charges, and terms of penalty. Most studies focus on label or task dependency among them to learn transferable representation for solving several tasks simultaneously. As all judgment subtasks can be viewed as classification problems, hard/soft parameters of the prediction model and representations (or some encoding layers) can be shared [34–36] by applying multi-task learning [8,13].

For example, based on text matching, Luo et al. [10] introduced a supervised attention module to solve charge prediction problems, and optimized charge and law article prediction tasks jointly. Based on the tree hierarchy, Wang et al. [12] proposed a hierarchical matching network to take advantage of the hierarchical structure between charges and law articles for crime classification. To simultaneously eliminate the ambiguity of similar law articles and fact descriptions, Lyu et al. [14] proposed a CEEN network for criminal element extraction and inputted the learned discriminative vector into the multi-task predictor.

Despite this, most methods assume that one task contributes equally to other tasks, without distinguishing different dependencies among tasks. To obtain deep semantic information on case facts, Yao et al. [11] introduced a gated hierarchical network to learn the dependencies among subtasks dynamically. To solve the problem of consistency and discriminative representation among subtasks, Dong and Niu [6] formalized LJP tasks as a node classification problem over the global consistency graph from training data.

Nevertheless, there exists a strict judgment order among subtasks, and the multi-task learning framework fails to reflect their ordered dependency. In response, Zhong et al. [7] first modeled this dependency using the semantic graph, and introduced a topological multi-task learning framework for dealing with these tasks together. To utilize result dependencies among subtasks, Yang et al. [5] further refine this kind of framework by adding backward propagation dependencies. Similarly, Yue et al. [17] proposed a circumstance-aware LJP framework to divide the fact description into two different parts, which are then used to predict other subtasks.

To address these issues, we propose HD-LJP for dealing with multi-task learning in judgment process. HD-LJP does not simply share the representation or model parameters between subtasks, but adopts relational distillation and logical dependency to better simulate the actual sentencing process. In other words, it can better model the label relationships, task dependencies, and decision order of subtasks, which can further significantly improves the scientific and reliability of judgment prediction.

3. Problem formalization

In this section, we briefly described a few basic definitions in this paper, such as legal judgment prediction, fact description, hierarchical semantics knowledge, label topological relation, and task judicial logic.

- **Legal Judgment Prediction (LJP):** It aims to predict the judgment result according to the input case facts and legal knowledge. As shown in Fig. 1, the input of LJP consists: case fact description D, legal text knowledge K, label topological relation L, and task judicial logic T. And, the output of LJP contains: law articles $\hat{y}_1$, charges $\hat{y}_2$, and terms of penalty $\hat{y}_3$.
- **Fact Description (D):** It is a part of the judgment document in Fig. 1(a), which mainly describes basic information about the defendant, crime times, crime locations, crime acts, crime tools, crime consequences, etc.
- **Hierarchical Semantics Knowledge (K):** It refers to the hierarchical semantic information contained in standard legal documents, such as article description and charge description in Fig. 1(b), which can assist human judges in deciding judgment results.
- **Label Topological Relation (L):** In established law articles, label relation of different LJP subtasks naturally exists. For example in Fig. 1(c), the charge "*Crime of Extortion*" is only related to the law article *274*.

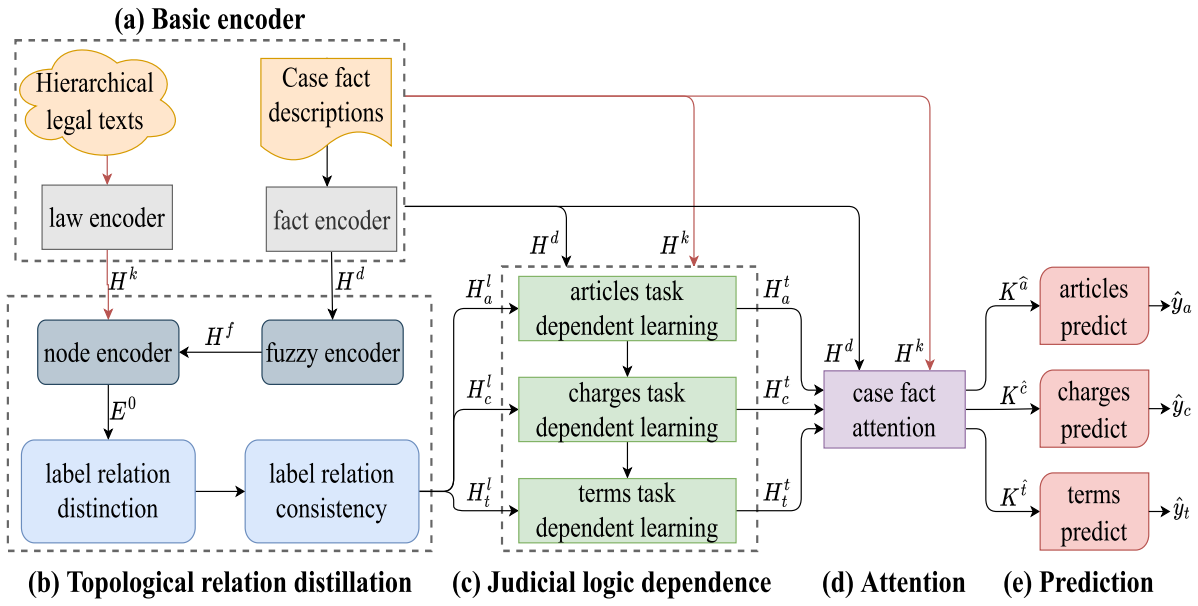


Fig. 2. The proposed framework: HD-LJP.

- **Task Judicial Logic (T):** There exists a strict logic order among three LJP subtasks, and their results influence and validate each other. As illustrated in Fig. 1(d), the judge first decide the applicable law articles, and charges and the terms of penalty are then determined in turn according to the above results.

4. Methodology

In this section, to realize the task of LJP, we introduce the hierarchical judgment prediction framework HD-LJP, and then present each module.

4.1. Proposed model framework

As shown in Fig. 2, we propose a joint learning framework (namely HD-LJP) based on hierarchical semantics knowledge, label topological relation, and task judicial logic for LJP. Specifically, the HD-LJP is mainly made up of four modules: (1) Basic Encoder Module; (2) Topological Relation Distillation Module; (3) Judicial Logic Dependence Module; (4) Case Fact Attention Module.

First of all, HD-LJP provides a unified representation of the hierarchical semantics knowledge contained in the established law articles and incorporates it into topological relation distillation and judicial logic dependence to make full use of the semantic information of case facts. Secondly, to model the label topological relation among multiple subtasks, HD-LJP employs consistency and distinction relation distillation to improve the differentiation of each subtask itself respectively. In addition, for simulating the task judicial logic in a real scenario, we utilize the results of intermediate subtasks to filter out the possible results of other subtasks and make auxiliary predictions. Further, to alleviate confusing verdict problems, we employ a fact-based attention mechanism to identify the fact words most closely related to the target label.

4.2. Basic encoder module

We design a basic document encoder to generate the unified vector representation for case fact description and legal text in the established law articles. Specifically, the basic encoder includes fact description encoder and legal text encoder.

Since Transformer [37] has achieved remarkable success in numerous fields, we first apply BERT [38], which is pre-trained on crime facts [39], to obtain the hidden representations H^d and h^d of the fact description. In detail, the word sequence of fact description $X^d = \{x_1^d, x_2^d, \ldots, x_{l_d}^d\}$ as the input, where l_d is the length of the word sequence. And, the context output for each token is represented as Eq. (1):

$$H^d = BERT(x_1^d, x_2^d, \ldots, x_{l_d}^d) \tag{1}$$

where $H^d = \{h_1^d, h_2^d, \ldots, h_{l_d}^d\} \in \mathbb{R}^{l_d \times d_h}$, d_h refers to the dimension of the final hidden layer.

Similarly, as for a legal text $X^k = \{x_1^k, x_2^k, \ldots, x_{l_k}^k\}$, we can obtain their hidden states $H^k = \{h_1^k, h_2^k, \ldots, h_{l_k}^k\} \in \mathbb{R}^{l_k \times d_h}$, where l_k denotes the length of the legal text.

In this way, $H^{k_a} \in \mathbb{R}^{m \times l_a \times d_h}$, $H^{k_c} \in \mathbb{R}^{n \times l_c \times d_h}$ and $H^{k_t} \in \mathbb{R}^{p \times l_t \times d_h}$ represent the knowledge vector of law article, charge, and term of penalty, where m, n, and p denote their respective legal number, l_a, l_c and l_t stand for the length in the established law articles, respectively.

$$H_a^k = mean(H^{k_a}) + max(H^{k_a}) \tag{2}$$

$$H_c^k = mean(H^{k_c}) + max(H^{k_c}) \tag{3}$$

$$H_t^k = mean(H^{k_t}) + max(H^{k_t}) \tag{4}$$

where $H_a^k \in \mathbb{R}^{m \times d_h}$, $H_c^k \in \mathbb{R}^{n \times d_h}$ and $H_t^k \in \mathbb{R}^{p \times d_h}$ represent the knowledge vector.

4.3. Topological relation distillation module

We employ a topological relation distillation module to model the label topological relation among multiple subtasks of LJP, and further improve the differentiation of each subtask itself respectively. Specifically, in Fig. 3 the topological relation distillation module includes four core components: fact fuzzy encoder, label node encoder, distinction relation distillation, and consistency relation distillation.

In the practical judgment process, human judges decide the outcome of criminal cases based on the circumstances of the crime, and different circumstances often lie in different parts of case facts [17]. Therefore, we utilize a fuzzy neural network (FNN) [40] to extract the circumstances related to law articles, charges, and terms of penalty, respectively. Taking the textual fact description $H^d = \{h_1^d, h_2^d, \ldots, h_{l_d}^d\} \in \mathbb{R}^{l_d \times d_h}$ of each case as input, we can obtain the fuzzy vector representation of each subtask as Eq. (5).

$$H_a^f, H_c^f, H_t^f = FNN(h_1^d, h_2^d, \ldots, h_{l_d}^d) \tag{5}$$

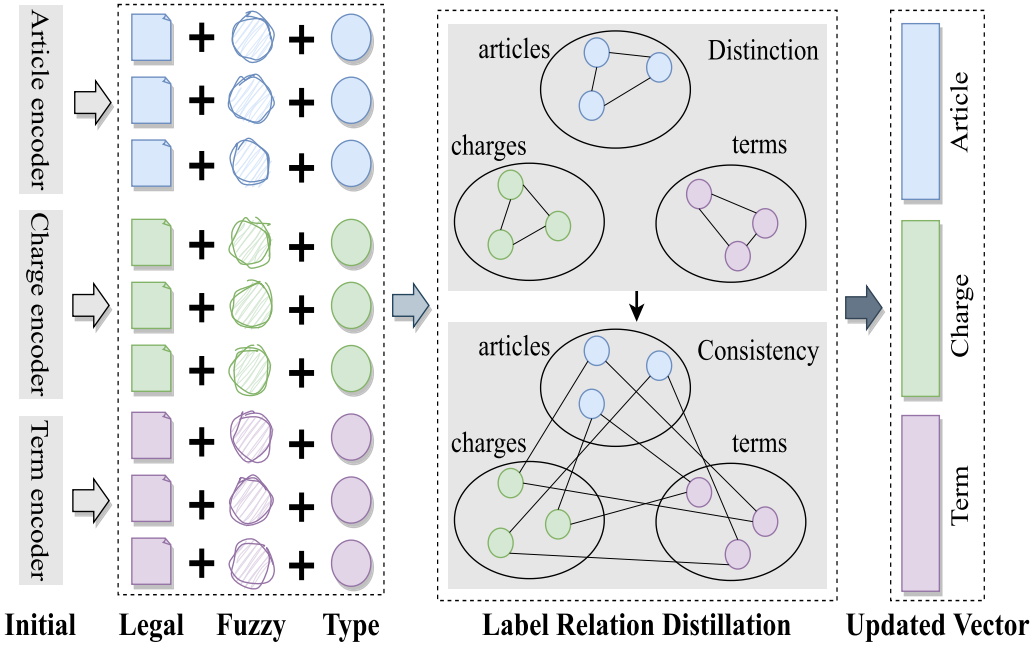


Fig. 3. Label topological relation distillation.

where $H_a^f \in \mathbb{R}^{m\times d_h}$, $H_c^f \in \mathbb{R}^{n\times d_h}$ and $H_t^f \in \mathbb{R}^{p\times d_h}$ represent the fuzzy vector of three subtasks.

Next, to better learn the label relation between subtasks, we initialize the node representation of three labels as Eq. (6), Eq. (7) and (8), which incorporates the knowledge representation based on legal text, the fuzzy representation based on fact description, and segment tokens representing each task.

$$E_a^0 = H_a^k + H_a^f + I_a \tag{6}$$

$$E_c^0 = H_c^k + H_c^f + I_c \tag{7}$$

$$E_t^0 = H_t^k + H_t^f + I_t \tag{8}$$

where $E_a^0 \in \mathbb{R}^{m\times d_h}$, $E_c^0 \in \mathbb{R}^{n\times d_h}$ and $E_t^0 \in \mathbb{R}^{p\times d_h}$ represent the final node vector, I_a, I_c and I_t represent the randomly initialized embedding of their token type, respectively.

Based on the above representation, inspired by [13], we employ the cosine similarity as Eq. (9) to calculate the correlation weight between two labels, and remove the edge with weights less than the threshold μ for constructing a label relation graph G_D inside a subtask as Eq. (10).

$$sim(i,j) = \frac{E_i^0 \cdot E_j^0}{\left|E_i^0\right| \cdot \left|E_j^0\right|} \tag{9}$$

$$M_s(i,j) = \begin{cases} 1 & i \neq j, sim(i,j) \geq \mu \\ 0 & otherwise \end{cases} \tag{10}$$

where $M_s(i,j) \in \mathbb{R}^{(m+n+p)\times(m+n+p)}$ represents the adjacency matrix inside the subtask.

To learn the node representation of the same class of subtasks, inspired by [13,16,17], we apply a distinction distillation mechanism. Based on the adjacency matrix M_s, similar feature are derived for nodes of the same task, and then utilize the distinct node to represent the central node as Eq. (11).

$$H^l = W_{\alpha_1} \cdot E^{l-1} - \frac{M_s \cdot E^{l-1} \cdot W_{\alpha_2}}{\left|N_d\right|} \tag{11}$$

where E^{l-1} is the representation of the node at $(l-1)_{th}$ layer, and $E^0 = \{E_a^0, E_c^0, E_t^0\} \in \mathbb{R}^{(m+n+p)\times d_h}$. $W_{\alpha_1} \in \mathbb{R}^{(m+n+p)\times(m+n+p)}$ and $W_{\alpha_2} \in \mathbb{R}^{d_h\times d_h}$ are the trainable parameter, $\left|N_d\right|$ is the number of edges in the label relation graph G_D.

In addition, to make the node representation consistent across different decision subtasks, inspired by [6], we construct a consistent graph G_C (as shown in Fig. 3) based on the label relationships that naturally exist between different judgment tasks in established legal text. For example in Fig. 1(c), the charge "*Crime of Extortion*" is only related to the law article *274*. Through this naturally occurring relationship, the node representation of the law article can be updated through its related charge representations (as Eq. (13)) according to the adjacency matrix as Eq. (12).

$$M_l(i,j) = \begin{cases} 1 & i \neq j, source(i) = source(j) \\ 0 & otherwise \end{cases} \tag{12}$$

$$H^{l+1} = W_{\beta_1} \cdot H^l + \frac{M_l \cdot H^l \cdot W_{\beta_2}}{\left|N_c\right|} \tag{13}$$

where $M_l(i,j) \in \mathbb{R}^{(m+n+p)\times(m+n+p)}$ represents the adjacency matrix between subtasks, and $source(i) = source(j)$ denotes the labels for subtask i and j come from the same legal text. $W_{\beta_1} \in \mathbb{R}^{(m+n+p)\times(m+n+p)}$ and $W_{\beta_2} \in \mathbb{R}^{d_h\times d_h}$ are the trainable parameter, $\left|N_c\right|$ is the number of edges in the label relation graph G_C. $H^l = \{H_a^l, H_c^l, H_t^l\} \in \mathbb{R}^{(m+n+p)\times d_h}$, and $H_a^l \in \mathbb{R}^{m\times d_h}$, $H_c^l \in \mathbb{R}^{n\times d_h}$ and $H_t^l \in \mathbb{R}^{p\times d_h}$ represent the final legal vector after distillation.

4.4. Judicial logic dependence module

To further improve the prediction performance of LJP tasks, we introduce the judicial logic dependence module to simulate the task judicial logic and order in a real scenario, which utilizes the results of intermediate subtasks to make auxiliary predictions of other subtasks. As shown in Fig. 4, the judicial logic dependence module includes three core components: law article-dependent learning, charge-dependent learning, and term-dependent learning.

Given the article description in the articles library, we apply the mean and max pooling operator to obtain the pooled article representation H_a^k, as Eq. (2). Then, we calculate the relevance score between law articles H_a^k and fact description H^d that indicates which law article words are most relevant to fact words as Eq. (14).

$$Score_a = H^d W_a H_a^{k^T} \tag{14}$$

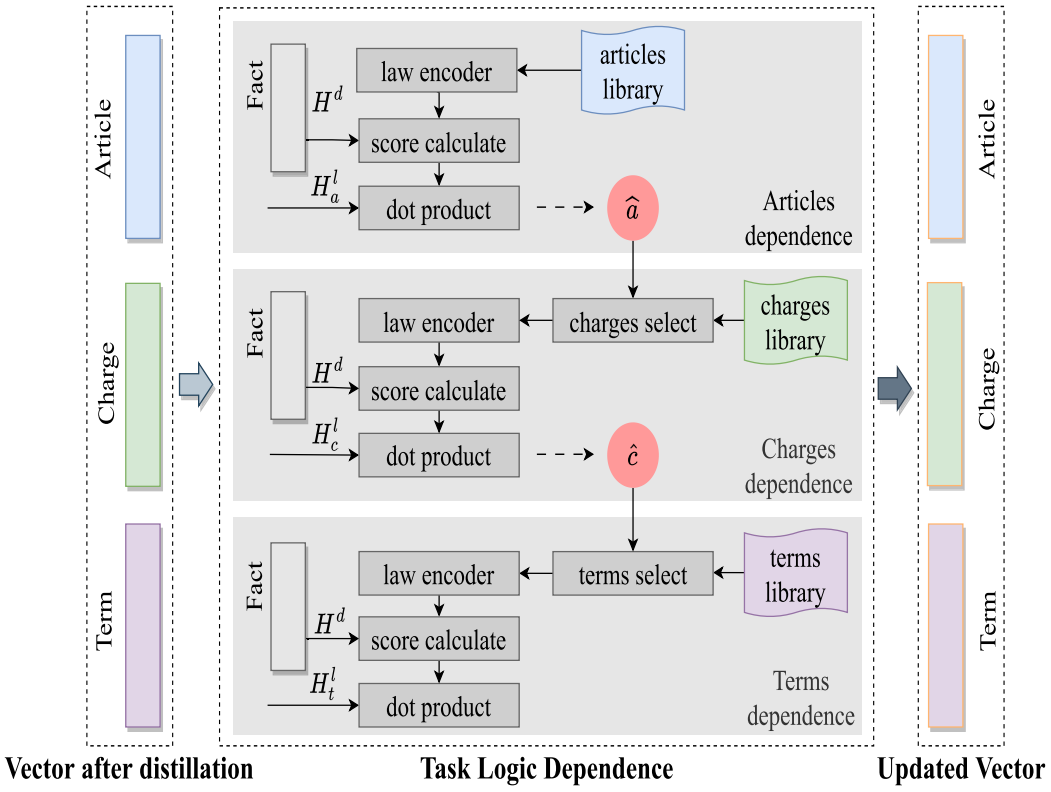


Fig. 4. Task judicial logic dependence.

Algorithm 1 Hierarchical Dependency-based Legal Judgment Prediction (HD-LJP)

Input: fact description $X^d = \{x_1^d, x_2^d, \ldots, x_{l_d}^d\}$; $X^k = \{x_1^k, x_2^k, \ldots, x_{l_k}^k\}$.
Output: prediction results $\hat{a}$, $\hat{c}$ and $\hat{t}$.
1: Initialize the representation of case fact H^d, based on Eq. (1);
2: Initialize the representation of legal text, H_a^k, H_c^k and H_t^k, based on Eq. (2), (3), (4);
3: Construct the label relation graph G_D and G_C, based on Eq. (10), (12);
4: Initialize the node representation, E_a^0, E_c^0 and E_t^0, based on Eq. (6), (7), (8);
5: **while** not coverage **do**
6: Obtain the node representation H_a^l, H_c^l and H_t^l by the topological Relation distillation Eq. (11), Eq. (13);
7: Obtain the feature representation H_a^t, H_c^t and H_t^t by the judicial logic dependent learning Eq. (15);
8: Update the fact representation $K^{\hat{a}}$, $K^{\hat{c}}$ and $K^{\hat{t}}$, based on case fact attention Eq. (16);
9: Prediction the probability distribution $\hat{y}_a$, $\hat{y}_c$ and $\hat{y}_t$, based on Eq. (20).
10: **end while**

where $W_a \in \mathbb{R}^{d_h \times d_h}$ is the trainable parameter, and we apply the dot product to obtain $H_a^t \in \mathbb{R}^{m \times d_h}$ as Eq. (15), which contains the attended law article vectors for the case description.

$$H_a^t = softmax(Score_a) \odot H_a^l \tag{15}$$

Different from the law article dependent learning, we first utilize the law article prediction result $\hat{a}$ (As shown in Fig. 2, articles prediction, we will describe in detail in Section 4.6) to filter the charge in the charges library, and screen out the most likely charge prediction results under the current facts. Then, we apply Eq. (3) to obtain the filtered charge representation $\widetilde{H}_c^k$, and calculate the charge representations $H_c^t \in \mathbb{R}^{\dot{a} \times d_h}$ ($\dot{a} \ll n$) that depend on law articles.

Similarly, in term-dependent learning, we can filter out the term of penalty in terms library based on the charge prediction result $\hat{c}$ (As shown in Fig. 2, charges prediction). Then, we apply Eq. (4) to obtain the filtered term representation $\widetilde{H}_t^k$, and calculate the term representations $H_t^t \in \mathbb{R}^{\dot{p} \times d_h}$ ($\dot{p} \ll p$) that depend on charges.

4.5. Case fact attention module

To alleviate confusing verdict problems, inspired by [17], we adopt a fact-based attention mechanism to recognize the fact words that are most closely related to the target label. For law article labels, as Eq. (16), we first obtain the vector $H_a^{d_k}$ according to the law article feature H_a^k, which can mitigate the loss of label semantics, and $H_a^{d_t}$ based on the dependent article features H_a^t.

$$H_a^{d_k} = Score_k H^d, H_a^{d_t} = Score_t H^d \tag{16}$$

where $Score_k \in \mathbb{R}^{l_d}$ and $Score_t \in \mathbb{R}^{l_d}$ are attention scores from the original feature representation and the dependent one as Eqs. (17) and (18), respectively.

$$Score_k = softmax(max_{col}(H^d W_k H_a^{k^T})) \tag{17}$$

$$Score_t = softmax(max_{col}(H^d W_t H_a^{t^T})) \tag{18}$$

Table 1
The statistics of the experimental datasets.

Datasets	# Training sets	# Test sets	# Validation sets	# Law articles	# charges	# Terms of penalty
CAIL-small	101,619	26,749	12,904	103	119	11
CAIL-big	1,587,979	185,120	–	118	130	11

where $W_k \in \mathbb{R}^{d_h \times d_h}$ and $W_t \in \mathbb{R}^{d_h \times d_h}$ are the trainable parameters, and max_col indicates the maximum value by column. Based on this, we can obtain article label aware fact representation $K^{\hat{a}} = [H^d, H_a^{d_k}, H_a^{d_t}] \in \mathbb{R}^{3d_h}$. Likewise, we can get $K^{\hat{c}} = [H^d, H_c^{d_k}, H_c^{d_t}] \in \mathbb{R}^{3d_h}$ for charge prediction, and $K^{\hat{t}} = [H^d, H_t^{d_k}, H_t^{d_t}] \in \mathbb{R}^{3d_h}$ for terms prediction.

4.6. Prediction and training

Here, at the prediction layer, we make judgment predictions for three LJP subtasks. Based on the article label aware representation of case fact $K^{\hat{a}}$, we first apply the gated recurrent unit (GRU) layer to encode the applicable law article as Eq. (19). Then, we apply a fully connected layer to learn the prediction probability $\hat{y}_a$ of law article as Eq. (20).

$$H^{\hat{a}} = GRU(K^{\hat{a}}) \tag{19}$$

$$\hat{y}_a = softmax(W_1 H^{\hat{a}} + b_1) \tag{20}$$

Where $\hat{a} = argmax(\hat{y}_{a_j})$, $\hat{y}_{a_j} \in \hat{y}_a$, $W_1 \in \mathbb{R}^{|y_a| \times d_h}$ and $b_1 \in \mathbb{R}^{d_h}$ represent the weight matrix and bias vector. And $|y_a|$ is the total count of law article labels. Similarly, we can get the probability distribution $\hat{y}_c$ and $\hat{y}_t$ for charges and the terms of penalty.

Finally, we employ the cross-entropy loss function [41] to train this model, and it takes the weighted sum as a training total loss by Eq. (21):

$$Loss = -\sum_{i=1}^{3} \lambda_i \sum_{j=1}^{|Y_i|} y_{i,j} log(\hat{y}_{i,j}) \tag{21}$$

where λ_i is the weight factor for each subtask, and $|Y_i|$ denotes the number of labels for subtask i. The complete process of our HD-LJP is summarized in Algorithm 1.

5. Experiments

In this section, to demonstrate the effectiveness of HD-LJP, We first introduce two training datasets, data processing methods, and provide the necessary training parameters of our model. Then, we compare our HD-LJP with some baselines on two CAIL-2018 datasets, illustrate the effect of each component module, and make some other model analyses.

5.1. Datasets description

We conduct our experiments on CAIL-2018 public datasets [18], which consist of CAIL-small and CAIL-big. These datasets consist of cases that contain fact descriptions, law articles, charges, and the terms of penalty. As for data processing, inspired by Zhong et al. [7], we conduct the following data preprocessing operations: (1) we first filter out some short text samples with fewer than 10 meaningful words in fact descriptions. (2) Filter out law articles and charges that have a minimal count of 100 cases. (3) Divide the terms of the penalty into 11 non-overlapping intervals, that is 11 categories. (4) Filter out the case samples with multiple law articles and charges. The detailed description of two datasets is shown in Table 1.

5.2. Baseline methods

To evaluate the prediction performance of the proposed hierarchical LJP method in this paper, we compare our method HD-LJP with the following baselines:

- **BERT** [42] obtains deep bidirectional representations from an unlabeled text by joint conditioning on both the left and right context in all layers.
- **FLA** [10] models the relation between case facts and judgment results of LJP by an attention-based neural network.
- **CNN** [43] extracts the feature information by utilizing multiple filters, and realize the text classification based on a convolutional neural network.
- **HARNN** [44] constructs a text feature extractor for document classification by a hierarchical attention-based RNN.
- **Few-Shot** [25] utilizes ten predefined attributes to enforce the fact representation, and then discriminate confusing charges.
- **TOPJUDGE** [7] is a multi-task learning framework, which integrates judgment subtasks and Directed Acyclic Graph (DAG) dependency into LJP.
- **MPBFN-WCA** [5] utilizes result dependencies among multiple subtasks by multi-perspective forward prediction and backward verification, and integrates word collocations features of fact descriptions to distinguish similar cases.
- **LADAN** [13] employs a distillation operator to automatically learn subtle differences between law articles for dealing with the confusing charges.
- **NeurlJudge+** [17] incorporates the results of intermediate subtasks and the label semantics to separate the case facts for making predictions for other subtasks.

5.3. Experimental setup

For the initial embedding of documents, we employ the pre-trained model which was trained on Chinese crime facts [39], and set the maximum token length of fact description to 512. For law article description, charge concept, and penalty description, Their dimensions are 150, 100, and 100, respectively. For the total loss, the weights λ_i are set as 1, the threshold μ as 0.3, and we use the training setup from the original paper for baseline methods. In the training process, we adopt the Adam optimizer [45], and the learning rate is 10^{-4}. We utilized PyTorch [46] to train every model for 16 epochs, the batch size is 128, and evaluate on test dataset.

Finally, following the work [6], we employ accuracy (Acc), macro-precision (MP), macro-recall (MR), and macro-F1 (F1) to evaluate the performance of the HD-LJP model on CAIL-small and CAIL-big datasets.

5.4. Overall results

In this subsection, based on the experiment results, we compare our HD-LJP with other baselines on law articles, charges, and terms of penalty tasks. The detailed results are shown in Tables 2 and 3. Specifically, we verify the effectiveness of the HD-LJP model from: (1) the overall prediction performance; (2) multi-task learning comparison; (3) task dependency comparison; (4) label dependency comparison; (5) data quantity and task type comparison.

As a whole, it can be found that our proposed HD-LJP consistently yields the best results for all the evaluation metrics on three LJP subtasks. Compared with the best baseline LADAN in Table 2, HD-LJP's F1 improvement on CAIL-small is 1.2%, 2.4%, and 13.4% for the article, charge, and term prediction task separately. Likewise, in Table 3, HD-LJP's F1 improvement is 2.2%, 1.2%, and 6.9% on CAIL-big separately. For dealing with the confusing law articles, LADAN proposed an attention mechanism to extract the key feature, but treated

Table 2
Judgment prediction performance on CAIL-small and the best model results are in bold.

Methods	Law articles				Charges				Terms of penalty			
	Acc	MP	MR	F1	Acc	MP	MR	F1	Acc	MP	MR	F1
BERT	79.14	78.77	73.26	72.63	82.24	83.55	78.68	78.83	38.41	33.96	28.71	28.43
CNN	78.71	76.02	74.87	73.79	82.41	81.51	79.34	79.61	35.40	33.07	29.26	29.86
HARNN	79.79	75.26	76.79	74.90	83.80	82.44	82.78	82.12	36.17	34.66	31.26	31.40
FLA	77.74	75.32	74.36	72.93	80.90	79.25	77.61	76.94	36.48	30.94	28.40	28.00
Few-Shot	79.30	77.80	77.59	76.09	83.65	80.84	82.01	81.55	36.52	35.07	26.88	27.14
TOPJUDGE	79.88	**79.77**	73.67	73.60	82.10	83.60	78.42	79.05	36.29	34.73	32.73	29.43
MPBFN-WCA	79.12	76.30	76.02	74.78	82.14	82.28	80.72	80.72	36.02	31.94	28.60	29.85
LADAN	81.20	78.24	77.38	76.47	85.07	83.42	82.52	82.74	38.29	36.16	32.49	32.65
NeurlJudge+	79.25	76.55	75.79	74.41	83.50	82.09	81.09	80.64	39.56	37.39	32.01	30.70
HD-LJP	**81.47**	79.62	**78.26**	**77.42**	**87.41**	**86.08**	**84.56**	**84.72**	**42.46**	**40.20**	**36.67**	**37.01**

Table 3
Judgment prediction performance on CAIL-big and the best model results are in bold.

Methods	Law articles				Charges				Terms of penalty			
	Acc	MP	MR	F1	Acc	MP	MR	F1	Acc	MP	MR	F1
BERT	93.54	82.65	64.66	68.97	93.02	84.11	67.27	71.96	52.38	41.75	32.90	33.71
CNN	95.84	83.20	75.31	77.47	95.74	86.49	79.00	81.37	55.43	45.13	38.85	39.89
HARNN	95.63	81.48	74.57	77.13	95.58	85.59	79.55	81.88	57.38	43.50	40.79	42.00
FLA	93.23	72.78	64.30	66.56	92.76	76.35	68.48	70.74	57.63	48.93	45.00	46.54
Few-Shot	96.12	85.43	80.07	81.49	96.04	88.30	80.46	83.88	57.84	47.27	42.55	43.44
TOPJUDGE	95.85	84.84	74.53	77.50	95.78	86.46	78.51	81.33	57.34	47.32	42.77	44.05
MPBFN-WCA	96.06	85.25	74.82	78.36	95.98	89.16	79.73	83.20	58.14	45.86	39.07	41.39
LADAN	96.57	86.22	80.78	82.36	96.45	88.51	83.73	85.35	59.66	51.78	45.34	46.93
NeurlJudge+	95.61	83.97	74.73	77.44	94.88	84.63	75.40	78.30	56.33	46.57	40.68	41.99
HD-LJP	**96.81**	**88.68**	**82.08**	**84.19**	**96.64**	**90.41**	**84.34**	**86.35**	**60.35**	**52.66**	**50.04**	**50.17**

each prediction subtask equally, which is unreasonable in real scenarios. By contrast, our HD-LJP not only considers label relationships but also introduces task dependencies to simulate real decision processes.

To evaluate the multi-task learning, following [13,14], CNN, HARNN, FLA, and Few-Shot were trained by utilizing the multi-task learning framework, and the final hidden state representation of the first token [CLS] in BERT is input to the multi-task predictor to predict the outcome of judgment. Compared with typical classification methods (i.e. CNN and HARNN), HD-LJP significantly improves the judgment performance on two datasets, and outperforms BERT on all LJP subtasks. FLA has poor performance because of its two-stage model, which can easily lead to the propagation of error verdicts. And, the ten predefined attributes are limited to charges, which makes Few-Shot unable to adapt to other subtasks. As for other LJP methods, they focus on either task dependence (TOPJUDGE and MPBFN-WCA) or label dependence (LADAN and NeurlJudge+), and the predictive performance is lower than HD-LJP, which indicates that joint learning of label and task dependency is extremely helpful for legal judgment prediction.

Task dependencies are often used to model the judgment order among subtasks, and it will have a negative impact on the judgment result if the pre-defined order is not reasonable. By comparing MPBFN-WCA and TOPJUDGE with other baselines, their task-dependent approach does not seem to be working for accuracy improvement. Only TOPJUDGE has a 0.1% higher accuracy rate than HARNN on law articles prediction. Different from them, HD-LJP adds label filtering to screen out possible labels of other subtasks in task-dependent learning, and the prediction performance on each subtask has been significantly improved. Compared with TOPJUDGE, HD-LJP's F1 score improvement on CAIL-small is 5.2%, 7.2%, and 25.8% for three LJP subtasks, and it is 8.6%, 6.2% and 13.9% on CAIL-big. Likewise, compared with MPBFN-WCA, the F1 score improvement on CAIL-small is 3.5%, 5.0%, and 24.0%, and it is 7.4%, 3.8% and 21.2% on CAIL-big.

Label dependency learning methods (i.e. LADAN and NeurlJudge+) take into account the popularity of confusing labels, and they have better performance than no-dependency learning models. Compared with HARNN, LADAN's F1 score improvement on CAIL-small is 2.1%, 0.8%, and 4.0% for three LJP subtasks separately, and it is 6.8%, 4.2%, and 11.7% on CAIL-big. Compared with the above task-dependent learning methods, LADAN's F1 score performance is at least 2.3% and 2.6% higher on two datasets respectively. Inspired by this, we introduce a topological relation distillation module to model label dependence among multiple subtasks. The experimental results also illustrate the effectiveness of our method. Compared with LADAN, our HD-LJP's F1 has improved by at least 1.2% on various subtasks.

As for data quantity and task type comparison, it can be seen that all models have better performance on CAIL-big, and the term of penalty has the worst predictive result compared with the other two subtasks. The number of training sets for CAIL-big is 1587979, while CAIL-small is 101619, a difference of nearly 15 times and an accuracy improvement of at least 10.6% for all subtasks. This verifies the importance of sufficient data in the training process. Compared to law articles and charges, sentences may be influenced by more factors, such as the results of intermediate subtasks, the word feature distribution in fact description, and semantic information about numbers. How to utilize these factors remains a challenging work, resulting in the failure of all models to the term of penalty effectively. Nevertheless, compared to the best baseline LADAN, our model still achieves optimal prediction performance, and its F1 has increased by 13.4% and 6.9% on two datasets.

5.5. Ablation experiment

We conduct ablation experiments on the CAIL-small, as shown in Table 4. Our HD-LJP incorporates three important auxiliary information (i.e. task logic order, label topological relation, and hierarchical semantic knowledge) for more accurate legal judgment prediction, they play different roles in performance improvement. Thus, to illustrate the effects of different parts in our HD-LJP, we consider seven model variants:

- **-Kno**: to prove the guiding significance of hierarchical semantic knowledge for LJP, we establish an HD-LJP model without legal text embedding.

Table 4
Ablation experiments performance on the CAIL-small dataset.

Methods	Law articles				Charges				Terms of penalty			
	Acc	MP	MR	F1	Acc	MP	MR	F1	Acc	MP	MR	F1
HD-LJP	**81.47**	**79.62**	**78.26**	**77.42**	**87.41**	**86.08**	**84.56**	**84.72**	**42.46**	**40.20**	**36.67**	**37.01**
-Kno	74.19	66.89	73.17	67.10	79.71	73.59	79.43	74.24	29.91	29.54	33.83	27.52
-Ord	72.70	65.66	71.11	65.74	78.60	72.58	77.29	72.67	29.63	27.74	33.52	26.43
-Rel	75.28	69.21	74.34	68.76	80.43	74.53	79.71	74.95	30.98	29.00	34.50	28.11
-Kno-Ord	64.31	59.77	66.08	57.41	73.29	67.19	73.67	66.39	25.58	27.08	31.58	24.00
-Kno-Rel	65.61	60.80	67.48	59.07	73.94	67.33	73.86	66.59	26.66	28.37	31.55	24.25
-Ord-Rel	64.76	60.39	66.54	58.62	73.97	67.82	74.33	67.52	26.58	27.36	32.11	23.98
-Kno-Ord-Rel	63.06	58.69	65.08	56.51	71.39	63.84	70.43	63.79	25.33	24.48	31.54	22.70

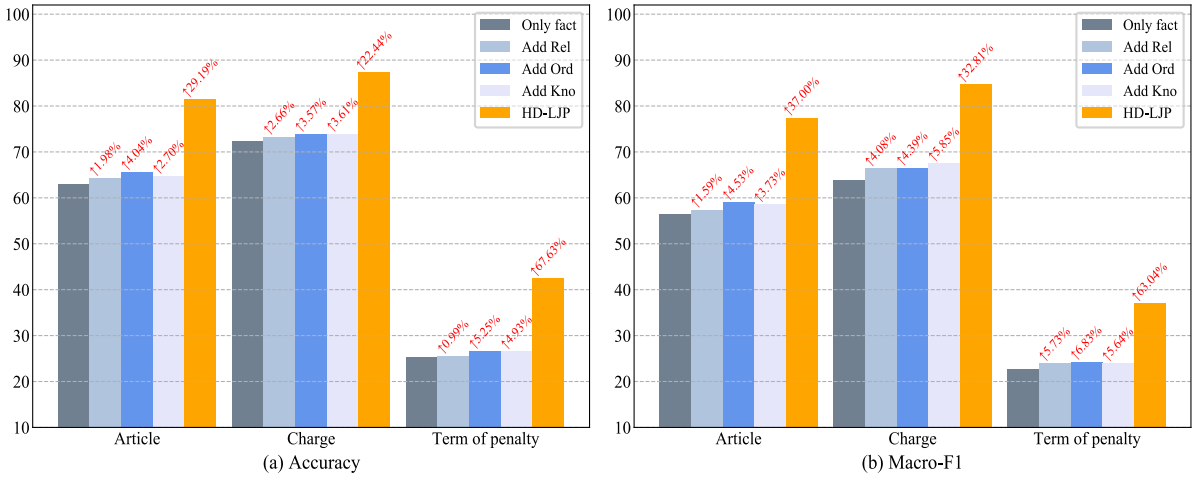


Fig. 5. Performance comparison of different auxiliary information.

- **-Ord**: to demonstrate the competence of task logic order, an HD-LJP model without the judicial logic dependence module is built.
- **-Rel**: to evaluate the significance of label topological relation, we remove the topological relation distillation module from HD-LJP.
- **-Kno-Ord**: we remove legal text embedding and the judicial logic dependence module from HD-LJP at the same time, that is, only the label topological relation is considered.
- **-Kno-Rel**: similarly, we remove legal text embedding and the topological relation distillation module from HD-LJP, that is, only the task logic order is considered.
- **-Ord-Rel**: this means that we only consider LJP methods that incorporate legal texts, which remove the judicial logic dependence and the topological relation distillation module.
- **-Kno-Ord-Rel**: the LJP method considers only the fact descriptions, without any external auxiliary information.

After removing a single module, the judgment performance of the corresponding model decreases, which confirms that integrating auxiliary knowledge can effectively improve the prediction. For example, on the three prediction subtasks of CAIL-small, the decline rates of "-Kno" are 15.38%, 14.12%, and 34.48% on the F1 score, respectively. The decline rates of "-Ord" are 17.77%, 16.58%, and 40.03% on the F1 score, and "-Rel" are 12.59%, 13.04%, and 31.66%.

In addition, as the number of eliminated modules increases, the prediction performance will gradually decrease, which can also illustrate the importance of the three auxiliary information. Indeed, the textual semantic knowledge contained in the established law articles can enrich the semantic information of case fact, and guide the process of judgment prediction. The logical order of subtasks can simulate the judicial logic in a real scenario, and ensure the rationality of the prediction process. The label topological relation can ensure the correct dependency among multiple subtasks, and further improve the accuracy of decision prediction.

Table 5
Model efficiency analysis based on training time and the number of parameters.

Models	# Parameters (million)	# Average time per epoch (min)	# Total time to reach stability (min)
Only fact	35	19	309
Add Kno	49	35	556
Add Rel	60	34	541
Add Ord	49	32	516
R-former	147	72	860
HD-LJP	**74**	**134**	**803**

5.6. Correlation analysis

Different types of auxiliary information contain different effective information, which focuses on different aspects of the decision prediction process, and thus the improvement of the prediction performance will be different. As shown in Fig. 5, based on the performance of "Only fact" (i.e. -Kno-Ord-Rel), "Add Ord" (adding task logic order, i.e. -Kno-Rel) improved F1 by 4.53%, 4.39%, and 6.83%, respectively. The improvement rates of "Add Kno" (adding hierarchical semantic knowledge, i.e. -Ord-Rel) are 3.73%, 5.85%, and 5.64% on F1, and "Add Rel" (adding label topological relation, i.e. -Kno-Ord) are 1.59%, 4.08%, and 5.73%. No one module has an absolute advantage in the three prediction tasks, which indicates that the three auxiliary information are all important, and we need to consider this information comprehensively in the process of decision prediction.

As observed above, we need to consider how to organically correlate different types of auxiliary information to maximize the LJP performance. On the one hand, our method utilizes the hierarchical semantic knowledge to initialize the node representation of the relation distillation module and to represent intermediate tasks in the logic-dependent process of the decision prediction. On the other hand, as for

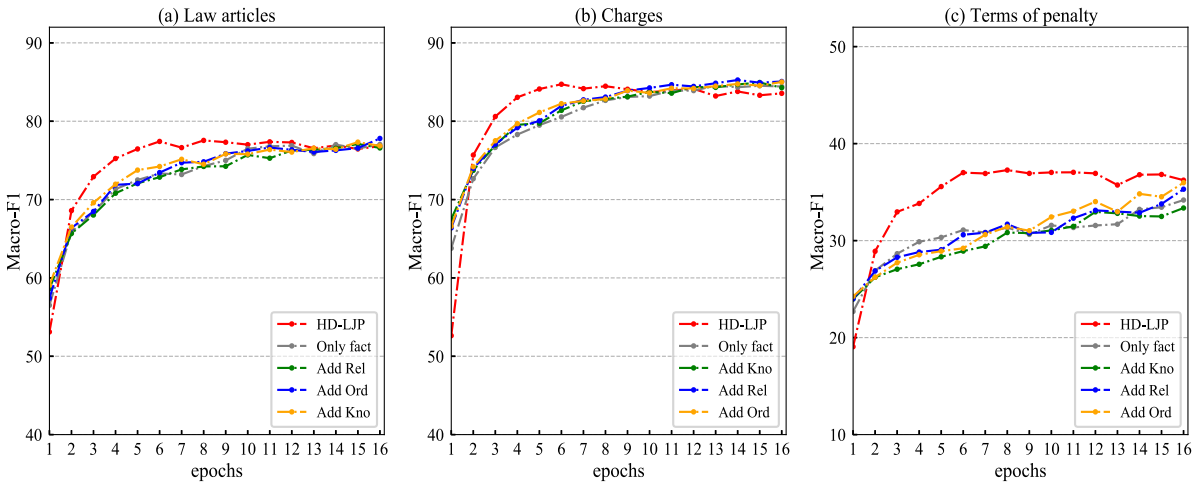


Fig. 6. Learning curves on different auxiliary information.

label topological relation and task logic order, we employ the output of the relational distillation module as the input of the label dependency module to filter out predictive labels, which can better fit the facts of the current case.

The ablation experiments also show that the integration of three kinds of auxiliary information can significantly improve the performance of three decision subtasks. For instance, the model (i.e. -Kno, -Ord, -Rel) that integrates two kinds of auxiliary information is better than the single information integrated model (i.e. -Kno-Ord, -Kno-Rel, -Ord-Rel) in Table 4, and their F1 score improves by at least 12.15%, 7.63%, and 10.13% on the three prediction tasks, respectively. The HD-LJP model that integrates three kinds of information has the best performance, model performance has improved by at least 12.59%, 13.03%, and 31.66% for the F1 score.

5.7. Efficiency analysis

To further illustrate the efficiency of the model that integrates auxiliary information, we study the training process of HD-LJP and its variants. To be specific, the learning curves with training epochs on three subtasks are demonstrated in Fig. 6, and a statistical analysis of the training time and the number of parameters for each model (including R-former) in Table 5. Based On this, we can come to the following conclusions.

In Fig. 6, as the training epochs increases, the macro-F1 score of HD-LJP and its variants improve continually, and then our model stabilizes. Notably, the growth rate of HD-LJP is significantly faster than that of other models, indicating that the proposed method does not lose training efficiency while integrating auxiliary information. Especially, when it comes to the terms of penalty prediction task, our HD-LJP significantly outperforms all variants at any epochs, which once again demonstrates the effectiveness of the HD-LJP.

Additionally, for a rigorous assessment of effectiveness, we display the average training time taken per epoch and the total time to reach stability for HD-LJP, its variants, and R-former, as well as the number of parameters for each model. In terms of the number of parameters (Table 5), due to the integration of functional modules of all variants, the number of parameters of HD-LJP has a certain increase compared with its variants, but the model performance has been significantly improved. Compared with the SOTA model R-former, the number of model parameters in HD-LJP is smaller and reduced by half. This illustrates that the model structure of our HD-LJP is more concise, and it will have faster model operation efficiency. In terms of model training time, although HD-LJP has the longest running time per epoch, it takes less time to reach a stable value than R-former. This also shows that the model training efficiency of the proposed method is better.

Undeniably, HD-LJP has a slight tendency to decline in the later stages of the training process (in Fig. 6), and each epoch takes too long during model training (in Table 5), which also prompts us to further optimize the model to integrate auxiliary information for more efficient legal judgment prediction.

5.8. Further case study

On the CAIL-2018 dataset, the class distribution of law articles and charge labels is extremely unbalanced, and confusing label relations are common in judgment results [6,13,17], which affect the overall performance of LJP. To investigate our HD-LJP's performance under this two circumstance, we test it on the cases of tail classes and confusing labels. The detailed description of the divided dataset is shown in Table 6.

Specifically, inspired by [47], we construct a long-tail dataset (namely CAIL-tail) based on law article and charge labels from CAIL-small, in which each label contains less than 200 cases; Its test set contains 72 law articles, 87 charges, and a total of 4087 test cases in Table 6. And, based on the confusing labels in [6,13,17,47], we utilize them to filter the test set of CAIL-small, and construct the CAIL-confusing, whose test set includes 6 law articles, 15 charges (such as *Crime of Forcible Seizure* and *Crime of Robbery*), and a total of 1529 test cases.

As shown in Table 6, both the divided datasets apply the same train and valid sets as CAIL-small in Table 1. That is, we train the model on the complete train set of CAIL-small, and test on the newly divided dataset. We compare the results with the existing state-of-the-art model R-former [6], which is also trained on the complete dataset (see Table 5) before evaluation. The experimental results are shown in Fig. 7, and it can be concluded from the above results that:

The prediction of tail classes is an essential and necessary ability in real judicial decision scenarios. Although R-former has achieved the best predictive performance in legal decisions compared to the current comparative baselines, its lack of consideration for long-tail labels prevented it from making effective predictions for this subset of cases. In contrast, our HD-LJP achieves better judgment results on both law article and charge prediction, which is partly related to the incorporation of legal knowledge into our model. Because the logical relations and semantic information implied in the established law articles can provide necessary knowledge guidance for the tail cases, so as to avoid wrong judgment results.

As for confusing label relations within LJP subtasks, R-former introduced the distinction distillation block for discriminative node representations. However, they did not do more experimental verification of this part, but simply focused on all possible label relationships to ensure

Table 6
The statistics of the divided dataset from CAIL-small. Law Articles, Charges and Terms of Penalty are from test cases. The train and valid sets are consistent with CAIL-small.

Datasets	# Train sets	# Valid sets	# Test cases	# Law articles	# Charges	# Terms of penalty
CAIL-tail	101,619	12,904	4,087	72	87	11
CAIL-confusing	101,619	12,904	1,529	6	15	11

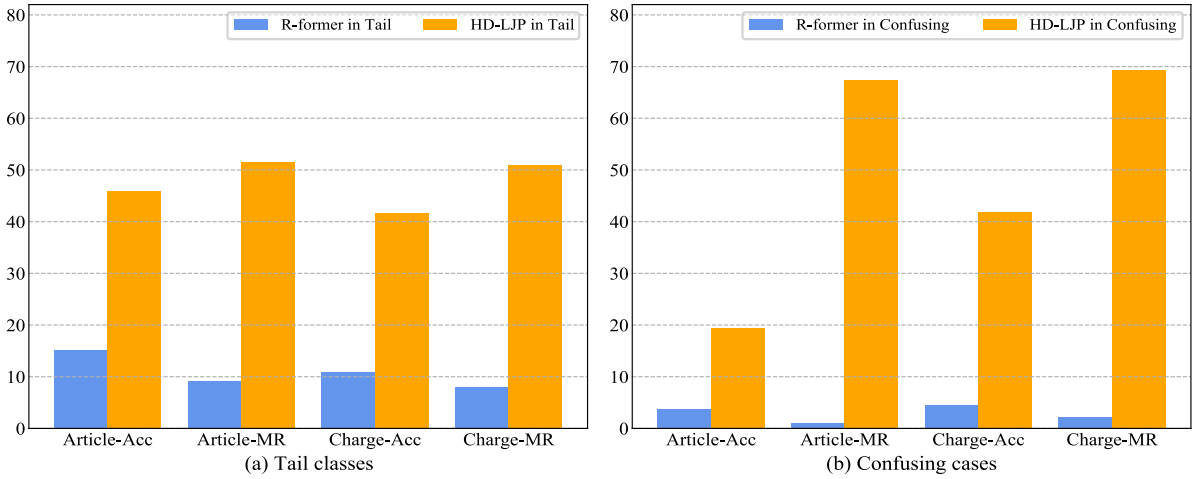


Fig. 7. Performance on the tail classes and confusing cases.

Fact Description	At 15:00 on December 4, 2013, the defendant Shi-A drove a motorcycle to take Shi-B. In front of the XXX tea shop, when the victim Su-A was unprepared, he snatched the handbag on Su-A's hand and drove away from the scene. The handbag contains RMB 800 yuan, an Apple 4S mobile phone and other properties. This phone was valued at RMB 3,237 yuan by the price department.	At 12:00 on October 13, 2013, the defendant Liu-A fled to XXX village in XXX District, XXX City. Later, he climbed the wall into the villager Li-A's home, and stole a pack of Yimeng Mountain brand cigarettes. In the process of theft, Liu-A was found by Li-A, and then he threatened Li-A with a knife. After that, Liu-A was caught by the crowd, when he was running away.
Ground Truth	***Article:*** 267 ***Charge:*** Crime of Forcible Seizure ***Prison Term:*** 18 months	***Article:*** 263 ***Charge:*** Crime of Robbery ***Prison Term:*** 120 months
R-former	***Article:*** 267 ✓ ***Charge:*** Crime of Forcible Seizure ✓ ***Prison Term:*** 24~36 months ✗	***Article:*** 267 ✗ ***Charge:*** Crime of Forcible Seizure ✗ ***Prison Term:*** 60~84 months ✗
HD-LJP	***Article:*** 267 ✓ ***Charge:*** Crime of Forcible Seizure ✓ ***Prison Term:*** 12~24 months ✓	***Article:*** 263 ✓ ***Charge:*** Crime of Robbery ✓ ***Prison Term:*** 84~122 months ✓

Fig. 8. A case study with specific intuitive examples (*Crime of Forcible Seizure* vs *Crime of Robbery*).

that the node representation was sufficiently discriminative. As can be seen from the results in Fig. 7, R-former has difficulty distinguishing confusing cases, and its prediction accuracy and recall rate is less than 10%. By contrast, our HD-LJP fully learns the consistency and distinction relations between LJP labels through relationship distillation, and simulates the logical relationships between subtasks with the help of the case attention mechanism. The experimental results also prove that the proposed method has better ability to distinguish confusing labels than R-former based on label dependence only.

In addition, we conduct a case study with specific intuitive examples (as shown in Fig. 8), which includes "*Crime of Forcible Seizure*" and "*Crime of Robbery*". They are both tail classes and confusing labels. As shown in Fig. 8, our HD-LJP can make correct judgments on these two cases. According to the criminal law, whoever carries a weapon to rob shall be convicted and punished in accordance with the provisions of the "*Crime of Robbery*". As a result, due to the lack of legal knowledge and task-dependent guidance, R-former determines both cases to be the "*Crime of Forcible Seizure*". It can be seen that legal knowledge, label dependence and task dependence are very effective for improving the prediction of judgment results, and the method in this paper combines them well.

6. Conclusions

In this work, we introduce a novel Hierarchical Dependency-based Legal Judgment Prediction framework (HD-LJP) for three subtasks of LJP, which integrate task judicial logic, label topological relation, and legal text semantic knowledge effectively. To make up for the lack of case fact information, HD-LJP incorporates legal texts into topological relation distillation and judicial logic dependence to take advantage of the semantic information. To model the label topological relation among multiple subtasks, HD-LJP employs consistency and distinction

relation distillation to improve the differentiation of each subtask itself respectively. To simulate the task judicial logic in a real scenario, we utilize the judgment result of intermediate tasks to filter out the possible results of other subtasks and make auxiliary predictions. Further, to alleviate confusing verdict problems, we adopt a fact-based attention mechanism to recognize the words most closely related to the target label. Compared to existing baseline methods, experimental results on two public LJP datasets validate the effectiveness of integrated learning in our HD-LJP.

In further research work, we will continue to explore the following directions: (1) how to more effectively integrate digital-related information in case fact to improve the penalty prediction performance. (2) we will further explore the evidence extraction methods to ensure the credibility and interpretability of judgment results.

CRediT authorship contribution statement

Yunong Zhang: Writing – review & editing, Writing – original draft, Validation, Conceptualization. **Xiao Wei:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Hang Yu:** Writing – review & editing, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research work was funded by the National Natural Science Foundation of China Youth Fund (No. 62302287).

References

[1] Y. Miao, F. Zhou, M. Pavlovski, W. Qian, Learning legal text representations via disentangling elements, Expert Syst. Appl. 249 (2024) 123749, http://dx.doi.org/10.1016/j.eswa.2024.123749.

[2] A.M. Moneus, Y. Sahari, Artificial intelligence and human translation: A contrastive study based on legal texts, Heliyon 10 (2024) e28106, http://dx.doi.org/10.1016/j.heliyon.2024.e28106.

[3] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, M. Sun, How does NLP benefit legal system: A summary of legal artificial intelligence, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5218–5230, http://dx.doi.org/10.48550/arXiv.2004.12158.

[4] Y.-X. Hong, C.-H. Chang, Improving colloquial case legal judgment prediction via abstractive text summarization, Comput. Law Secur. Rev. 51 (2023) 105863, http://dx.doi.org/10.1016/j.clsr.2023.105863.

[5] W. Yang, W. Jia, X. Zhou, Y. Luo, Legal judgment prediction via multi-perspective bi-feedback network, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 4085–4091, http://dx.doi.org/10.24963/ijcai.2019/567.

[6] Q. Dong, S. Niu, Legal judgment prediction via relational learning, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 983–992, http://dx.doi.org/10.1145/3404835.3462931.

[7] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, M. Sun, Legal judgment prediction via topological learning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3540–3549, http://dx.doi.org/10.18653/v1/D18-1390.

[8] S. Yang, S. Tong, G. Zhu, J. Cao, Y. Wang, Z. Xue, H. Sun, Y. Wen, MVE-FLK: A multi-task legal judgment prediction via multi-view encoder fusing legal keywords, Knowl.-Based Syst. 239 (2021) 107960, http://dx.doi.org/10.1016/j.knosys.2021.107960.

[9] Y. Liu, Y. Wu, Y. Zhang, C. Sun, W. Lu, F. Wu, K. Kuang, ML-LJP: Multi-law aware legal judgment prediction, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 1023–1034, http://dx.doi.org/10.1145/3539618.3591731.

[10] B. Luo, Y. Feng, J. Xu, X. Zhang, D. Zhao, Learning to predict charges for criminal cases with legal basis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2727–2736, http://dx.doi.org/10.18653/v1/D17-1289.

[11] F. Yao, X. Sun, H. Yu, Y. Yang, W. Zhang, K. Fu, Gated hierarchical multi-task learning network for judicial decision prediction, Neurocomputing 411 (2020) 313–326, http://dx.doi.org/10.1016/j.neucom.2020.05.018.

[12] P. Wang, Y. Fan, S. Niu, Z. Yang, Y. Zhang, J. Guo, Hierarchical matching network for crime classification, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 325–334, http://dx.doi.org/10.1145/3331184.3331223.

[13] N. Xu, P. Wang, L. Chen, L. Pan, X. Wang, J. Zhao, Distinguish confusing law articles for legal judgment prediction, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3086–3095, http://dx.doi.org/10.18653/v1/2020.acl-main.280.

[14] Y. Lyu, Z. Wang, Z. Ren, P. Ren, Z. Chen, X. Liu, Y. Li, H. Li, H. Song, Improving legal judgment prediction through reinforced criminal element extraction, Inf. Process. Manag. 59 (2022) 102780, http://dx.doi.org/10.1016/j.ipm.2021.102780.

[15] Q. Zhao, T. Gao, N. Guo, LA-MGFM: A legal judgment prediction method via sememe-enhanced graph neural networks and multi-graph fusion mechanism, Inf. Process. Manag. 60 (2023) 103455, http://dx.doi.org/10.1016/j.ipm.2023.103455.

[16] F. Yao, X. Sun, H. Yu, W. Zhang, K. Fu, Commonalities-, specificities-, and dependencies-enhanced multi-task learning network for judicial decision prediction, Neurocomputing 433 (2020) 169–180, http://dx.doi.org/10.1016/j.neucom.2020.10.010.

[17] L. Yue, Q. Liu, B. Jin, H. Wu, K. Zhang, Y. An, M. Cheng, B. Yin, D. Wu, 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, 2021, pp. 973–982, http://dx.doi.org/10.1145/3404835.3462826.

[18] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang, J. Xu, CAIL2018: A large-scale legal dataset for judgment prediction, 2018, http://dx.doi.org/10.48550/arXiv.1807.02478, arXiv, arXiv:1807.02478.

[19] C.-L. Liu, C.-T. Chang, J.-H. Ho, Case instance generation and refinement for case-based criminal summary judgments in Chinese, J. Inf. Sci. Eng. 20 (2004) 783–800.

[20] Y.-H. Liu, Y.-L. Chen, W.-L. Ho, Predicting associated statutes for legal problems, Inf. Process. Manag. 51 (2015) 194–211, http://dx.doi.org/10.1016/j.ipm.2014.07.003.

[21] O.M. Sulea, M. Zampieri, S. Malmasi, M. Vela, L.P. Dinu, J. van Genabith, Exploring the use of text classification in the legal domain, in: Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts, 2017, https://arxiv.org/abs/1710.09306.

[22] S. Bi, Z. Ali, T. Wu, G. Qi, Knowledge-enhanced model with dual-graph interaction for confusing legal charge prediction, Expert Syst. Appl. 249 (2024) 123626, http://dx.doi.org/10.1016/j.eswa.2024.123626.

[23] S. Tong, J. Yuan, P. Zhang, L. Li, Legal judgment prediction via graph boosting with constraints, Inf. Process. Manage. 61 (2024) 103663, http://dx.doi.org/10.1016/j.ipm.2024.103663.

[24] G. Feng, Y. Qin, R. Huang, Y. Chen, Criminal Action Graph: A semantic representation model of judgement documents for legal charge prediction, Inf. Process. Manage. 60 (2023) 103421, http://dx.doi.org/10.1016/j.ipm.2023.103421.

[25] Z. Hu, X. Li, C. Tu, Z. Liu, M. Sun, Few-shot charge prediction with discriminative legal attributes, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 487–498.

[26] J. Guo, Z. Liu, Z. Yu, Y. Huang, Y. Xiang, Few shot and confusing charges prediction with the auxiliary sentences of case, J. Softw. 32 (3139–3150) (2021) http://dx.doi.org/10.13328/j.cnki.jos.006028.

[27] Z. Xu, xia li, Y. Li, Z. Wang, Y. Fanxu, X. Lai, Multi-task legal judgement prediction combining a subtask of the seriousness of charges, in: China National Conference on Chinese Computational Linguistics, 2020, pp. 415–429, http://dx.doi.org/10.1007/978-3-030-63031-7_30.

[28] L. Gan, K. Kuang, Y. Yang, F. Wu, Judgment prediction via injecting legal knowledge into neural networks, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 12866–12874, http://dx.doi.org/10.1609/aaai.v35i14.17522.

[29] P. Wang, Z. Yang, S. Niu, Y. Zhang, L. Zhang, S. Niu, Modeling dynamic pairwise attention for crime classification over legal articles, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 485–494, http://dx.doi.org/10.1145/3209978.3210057.

[30] H. Zhong, J. Zhou, W. Qu, Y. Long, Y. Gu, An element-aware multi-representation model for law article prediction, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 6663–6668, http://dx.doi.org/10.18653/v1/2020.emnlp-main.540.

[31] H. Ye, X. Jiang, Z. Luo, W.-H. Chao, Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 1854–1864, http://dx.doi.org/10.18653/v1/N18-1168.

[32] H. Chen, D. Cai, W. Dai, Z. Dai, Y. Ding, 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, 2019, pp. 6362–6367, http://dx.doi.org/10.18653/v1/D19-1667.

[33] S. Long, C. Tu, Z. Liu, M. Sun, Automatic judgment prediction via legal reading comprehension, in: China National Conference on Chinese Computational Linguistics, 2018, pp. 558–572.

[34] Z. Zhang, W. Yu, M. Yu, Z. Guo, M. Jiang, A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods, 2022, arXiv, arXiv:2204.03508.

[35] C. Ruiz, C.M. Alaíz, J.R. Dorronsoro, A survey on kernel-based multi-task learning, Neurocomputing 577 (2024) 127255, http://dx.doi.org/10.1016/j.neucom.2024.127255.

[36] L. Wang, J. Peng, C. Zheng, T. Zhao, L. Zhu, A cross modal hierarchical fusion multimodal sentiment analysis method based on multi-task learning, Inf. Process. Manage. 61 (2024) 103675, http://dx.doi.org/10.1016/j.ipm.2024.103675.

[37] A. Vaswani, N.M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st Conference on Neural Information Processing Systems, NIPS, 2017, pp. 5999–6009, https://arxiv.org/abs/1706.03762.

[38] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, S. Wang, G. Hu, Pre-training with whole word masking for Chinese BERT, IEEE/ACM Trans. Audio Speech Lang. Process. 29 (2019) 3504–3514, http://dx.doi.org/10.48550/arXiv.1906.08101.

[39] H. Zhong, Z. Zhang, Z. Liu, M. Sun, Open Chinese Language Pre-trained Model Zoo, Tech. Rep., Tsinghua University, 2019, URL https://github.com/thunlp/openclap.

[40] B. Fang, C. Zheng, H. Wang, T. Yu, Two-stream fused fuzzy deep neural network for multiagent learning, IEEE Trans. Fuzzy Syst. 31 (2023) 511–520, http://dx.doi.org/10.1109/TFUZZ.2022.3214001.

[41] J.E. Shore, R.W. Johnson, Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy, IEEE Trans. Inform. Theory 26 (1980) 26–37, http://dx.doi.org/10.1109/TIT.1980.1056144.

[42] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186, http://dx.doi.org/10.18653/v1/N19-1423.

[43] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1746–1751, http://dx.doi.org/10.3115/v1/D14-1181.

[44] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E.H. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1480–1489, http://dx.doi.org/10.18653/v1/N16-1174.

[45] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations, 2014, http://dx.doi.org/10.48550/arXiv.1412.6980.

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Proceedings of the 33th Conference on Neural Information Processing Systems, 2019, pp. 8026–8037, http://dx.doi.org/10.48550/arXiv.1912.01703.

[47] H. Zhang, Z. Dou, Y. Zhu, J.-R. Wen, Contrastive learning for legal judgment prediction, ACM Trans. Inf. Syst. 41 (2023) 1–25, http://dx.doi.org/10.1145/3580489.