Legal Judgment Prediction based on Knowledge-enhanced Multi-Task and Multi-Label Text Classification

Ang Li[1*], **Yiquan Wu**[2,3*†], **Ming Cai**[1†], **Adam Jatowt**[4], **Xiang Zhou**[2]
Weiming Lu[1], **Changlong Sun**[5], **Fei Wu**[1], **Kun Kuang**[1†]

[1]College of Computer Science and Technology, Zhejiang Unirersity,
[2]Guanghua Law School, Zhejiang University, [3]Al&Law Lab, Zhejiang University,
[4]University of Innsbruck, Innsbruck, Austria, [5]Alibaba Group, Hangzhou, China
{leeyon, wuyiquan, cm, 0020355, luwm, wufei, kunkuang}@zju.edu.cn
adam.jatowt@uibk.ac.at, changlong.scl@taobao.com

Abstract

Legal judgment prediction (LJP) is an essential task for legal AI, aiming at predicting judgments based on the facts of a case. Legal judgments can involve multiple law articles and charges. Although recent methods in LJP have made notable progress, most are constrained to single-task settings (e.g., only predicting charges) or single-label settings (e.g., not accommodating cases with multiple charges), diverging from the complexities of real-world scenarios. In this paper, we address the challenge of predicting relevant law articles and charges within the framework of legal judgment prediction, treating it as a multi-task and multi-label text classification problem. We introduce a knowledge-enhanced approach, called K-LJP, that incorporates (i) "label-level knowledge" (such as definitions and relationships among labels) to enhance the representation of case facts for each task, and (ii) "task-level knowledge" (such as the alignment between law articles and corresponding charges) to improve task synergy. Comprehensive experiments demonstrate our method's effectiveness in comparison to state-of-the-art (SOTA) baselines.

1 Introduction

Due to the recent advancement of machine learning, various NLP techniques have been applied in Legal Artificial Intelligence (LegalAI) to assist judges in different ways, including controversy focus mining (Duan et al., 2019), legal document generation (Wu et al., 2020) and legal judgment prediction (Luo et al., 2017; Wu et al., 2022). As one of the most important tasks in LegalAI, legal judgment prediction (LJP) has been studied for quite a long time (Keown, 1980; Lin et al., 2012). Given the case's fact description, LJP aims to predict the likely judgment that includes law articles and charges. In the actual trial scene, as portrayed in Fig. 1, a case may involve several law articles and charges. However, most existing methods simplify the LJP to a single-label setting (Zhong et al., 2018; Yue et al., 2021a) (e.g., ignoring multi-label cases) or single-task setting (Wang et al., 2019) (e.g., predicting charges only), which is inconsistent with real-world scenarios.

To bridge the gap between the simplified settings and the real scenarios, we formulate the problem of predicting law articles and charges as a multi-task and multi-label text classification task and we utilize the corresponding legal knowledge. Based on this we consider two levels of legal knowledge: 1) **"Label-level knowledge" in each task:** All law articles and charges are defined with detailed definitions, and there are complex relationships among them (e.g., competing law articles). 2) **"Task-level knowledge" across tasks:** As Fig. 1 shows, law articles and charges are aligned in a many-to-many way. The alignment between law articles and charges can be represented as a mapping dictionary. However, the alignment is not strict (e.g., the multiple mapping charges of a law article are not all guaranteed to appear), which makes the two classification tasks linked with each other, yet without a means to be combined into a single task.

In this paper, we propose a novel knowledge-enhanced LJP method (K-LJP) to incorporate the above mentioned types of legal knowledge into the model. For the "label-level knowledge", in each task, label definitions are used to obtain the initial embedding of each label, and the Transformer decoder is retrofitted to learn the complex relationships among the labels and between the label and the case's fact simultaneously. For the "task-level knowledge", we design a special alignment loss that utilizes the alignment between the law articles and charges to synergize the tasks, and we construct two mapping dictionaries from the Chinese Code of Law for the alignment loss calculation.

To validate the efficacy of our proposed method,

* Equal contribution.
† Corresponding Author.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6957–6970
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

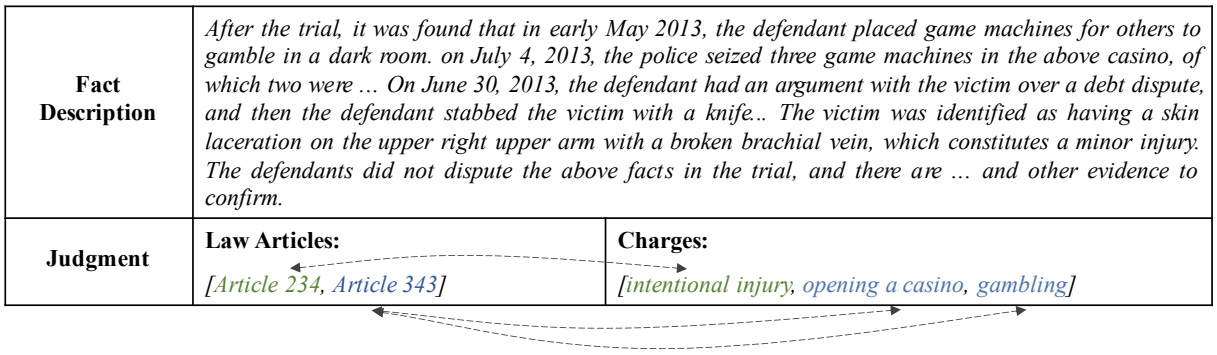

Figure 1: An example case that involves multiple law articles and multiple charges. The dotted lines refer to the alignment between the law articles and the charges.

we conducted experiments using the widely recognized Chinese legal dataset, CAIL2018. The experimental outcomes demonstrate that our model outperforms existing methods in both accuracy and alignment ratio metrics within a multi-label setting. Furthermore, even when the experimental context was simplified to a single-label setting, our method still exhibited superior performance compared to models specifically tailored for the single-label setting.

To sum up, we make the following contributions:

- We investigate the prediction of law articles and charges in legal judgment prediction as a multi-task and multi-label text classification problem, which, as we believe, is more practical in real scenarios.
- We propose a novel knowledge-enhanced multi-task and multi-label text classification method called K-LJP for the LJP task. In the K-LJP, the label-level knowledge (e.g., label definition and relationship) is used better to learn the representation of the case's fact thus enhancing each classification task, and the task-level knowledge (e.g., alignment between law articles and charges) is used to synergize the tasks.
- We carry out extensive experiments on a real world dataset, the results of which demonstrate the effectiveness of our K-LJP model. The ablation study highlights the benefit of knowledge injection.
- To support the reproducibility, we make the code publicly available[1].

[1] https://github.com/LIANG-star177/KLJP

2 Related Work

2.1 LegalAI

Legal Artificial Intelligence (LegalAI) aims to apply the technology of artificial intelligence to the legal domain. Recently, researchers have been exploring the methods to improve the efficiency of judges and lawyers from both civil-law countries (e.g., France, Germany) and common-law countries (e.g., the United Kingdom, the United States, India) (Cui et al., 2022). In general, many relevant tasks have been proposed, such as legal judgment prediction (Chalkidis et al., 2019; Xiao et al., 2021a), legal questions classification (Xiao et al., 2017), legal event detection (Yao et al., 2022), court view generation (Li et al., 2024b; Liu et al., 2024; Zhou et al., 2024) and so on. In this work, we focus on the task of legal judgment prediction, which is one of the fundamental tasks in LegalAI.

2.2 Legal Judgment Prediction

Legal judgment prediction (LJP), which already has been studied for decades, focuses on predicting legal judgment based on the case facts (Yue et al., 2021b; Xu et al., 2020; Feng et al., 2022a; Lyu et al., 2022; Zhao et al., 2022; Gan et al., 2021; Yang et al., 2019; Paul et al., 2021; Li et al., 2024a). In the early years, most LJP methods were rule-based, which require lots of manually extracted features (Keown, 1980). The rule-based methods are simple and efficient but are difficult to generalize because of the high cost of feature extraction. Recently, several deep learning methods have been applied for legal judgment prediction. Luo et al. (2017) incorporate the attention mechanism into the model to better represent the input. Zhong et al. (2018) propose a method to learn the topological relationship among the subtasks in the LJP. Liu

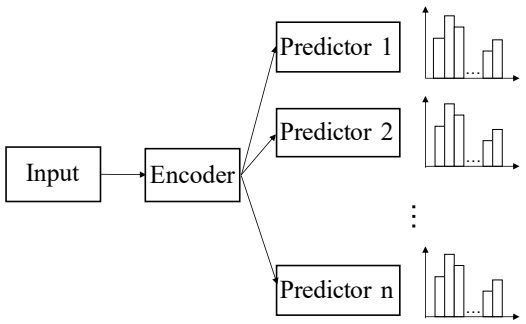

Figure 2: The vanilla model.

et al. (2022a) use contrastive case relations to augment LJP. Wu et al. (2022) generates the rationales before the prediction to get the interpretability. Yue et al. (2021a) divides the fact description into adjudging circumstance and sentencing circumstance and exploits them to make the prediction.

However, most existing methods focus on extracting efficient features from the input and simplifying the task into a single-label or single-task setting. In our work, we investigate the prediction of law articles and charges as a multi-task and multi-label text classification task, which is more relevant to the reality of legal practice.

3 Problem Formulation

We first provide the definitions as follows.

Fact description consists of several sentences, which describe the events relevant to a case. Here, we denote the fact description as $f = \{w_t^f\}_{t=1}^{l_f}$, where l_f is the number of words in a fact description.

Judgment includes several law articles, and several charges. The law articles and the charges are in the form of labels. We denote the law articles as $a = \{a_i\}_{t=1}^{l_a}$ and the charges as $c = \{c_i\}_{t=1}^{l_c}$, where l_a and l_c are the number of law articles and charges, respectively.[2]

Thus, the problem in our work is defined as:

Problem 1 (Legal Judgment Prediction)**.** *Given the fact description f, the task is to predict the judgment (a, c).*

4 Method

In this section, we describe the K-LJP method in detail. We begin by designing a vanilla model for the multi-task and multi-label text classification, and then introduce the proposed method which is an extended version of the vanilla model.

4.1 Vanilla Model

As Fig. 2 shows, the vanilla model consists of a shared encoder and a set of independent predictors, where each predictor corresponds to a given task.

4.1.1 Encoder

Given the input text sequence $f = \{w_t^f\}_{t=1}^{l_f}$, the encoder E aims to transform it to a sequence of hidden states $h^f \in \mathbb{R}^{l_f \times d}$:

$$h^f = \text{Encode}(f), \tag{1}$$

where d is the output dimension of the hidden state. The encoder E can be implemented using some common network structures such as CNN (LeCun et al., 1989), LSTM (Sutskever et al., 2014) or Transformer (Vaswani et al., 2017), or a pre-trained model like BERT (Devlin et al., 2019).

Then, an information aggregation operation (e.g., mean pooling or attention mechanism (Bahdanau et al., 2015)) is applied to h^f sequence to get the final representation of the input $h^{f*} \in \mathbb{R}^d$.

4.1.2 Predictors

We define predictors as $P = \{P_i\}_{i=1}^{n}$, where P_i is the predictor for the i-th task, and n is the number of tasks in multi-task classification. Here, each predictor in P has the same structure but aims to predict the labels of different tasks. For the predictor P_i, given the representation of the input h^{f*}, it outputs the predicted probability $\hat{y}_i \in \mathbb{R}^{l_i}$ of the i-th task, where l_i is the number of the labels in the i-th task, and the probability of the j-th label $\hat{y}_{ij}$ can be calculated as follows:

$$\hat{y}_{ij} = \text{sigmoid}(W_{ij}^T h^{f*}), \tag{2}$$

where $W_{ij} \in \mathbb{R}^d$ is the trainable parameter of the fully connected layer, and the sigmoid function is used to transfer the output value into a probability interval from 0 to 1.

Through the predictors P, every label in each task has its probability assigned.

4.1.3 Training and Inference

The cross-entropy loss has been proved suitable for multi-label text classification task (Nam et al., 2014). The loss function of the i-th predictor P_i is

[2] In our setting, we address the challenge of law articles and charges prediction, and we do not focus on predicting the penalty, which is considered as a downstream task and one that would need to be modelled as a regression problem.

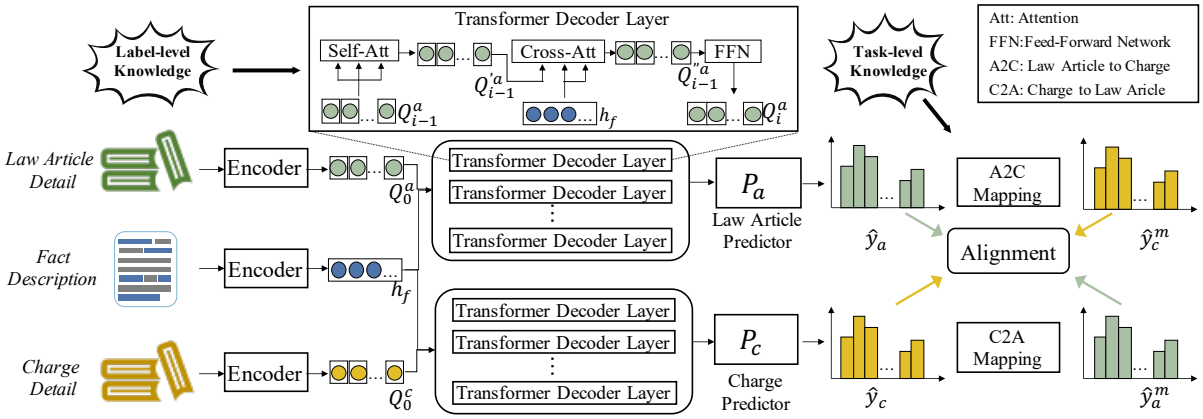

Figure 3: The architecture of K-LJP.

calculated as follow:

$$\mathcal{L}_i = -\sum_{j=1}^{l_i} \left(y_{ij} \log\left(\hat{y}_{ij}\right)\right) + \left(1 - y_{ij}\right) \log\left(1 - \hat{y}_{ij}\right), \quad (3)$$

where $\hat{y}_{ij} \in [0, 1]$ is the predicted probability, and $y_{ij} \in \{0, 1\}$ indicates the ground truth.

Then, the cross-entropy loss of the vanilla model can be summed up as $\mathcal{L}_C = \sum_{i=1}^{n} \mathcal{L}_i$, where n is the number of tasks.

In the inference, same as in Rios and Kavuluru (2018), we use the threshold method and set the threshold to 0.5, which means a label will be predicted when its probability is more than 0.5 [3].

4.2 K-LJP

In the LJP task, the input text sequence is the fact description. There are two predictors in P: law article predictor P_a and charge predictor P_c. As Fig. 3 shows, based on the vanilla model, we introduce the knowledge-enhanced method K-LJP. There are two main improvements: 1) First, we inject label-level knowledge into the encoder for each task; 2) Then, we propose an alignment loss to utilize the task-level knowledge such that the law articles and the charges are aligned.

4.2.1 Encoder

First, we do the same operation as the vanilla model does to obtain the sequence of hidden states $h^f \in \mathbb{R}^{l_f \times d}$ of the input. Then, since every law article and every charge has a detailed definition in the Code of Law, we use Transformer networks and a mean pooling operation to get the initial representation of each label in every task. Specifically, we define the initial label representations of law articles as $Q_0^a \in \mathbb{R}^{l_a \times d}$, and the initial label representations of charges as $Q_0^c \in \mathbb{R}^{l_c \times d}$, where l_a and l_c are the number of labels in each task.

Inspired by the image classification model Query2Label (Liu et al., 2021), we use the Transformer decoder (Vaswani et al., 2017) to learn the complex relationships among the labels and between the label and fact. Here, we use the law articles as example, while the same approach is done for the charges.

Given the initial label representations of a law article Q_0^a and fact representation h^f, a Transformer decoder layer mainly does three operations [4]:

1) **Self-Attention**:

$$Q_i'^a = \text{MultiHeadAtt}\left(Q_{i-1}^a, Q_{i-1}^a, Q_{i-1}^a\right); \quad (4)$$

2) **Cross-Attention**:

$$Q_i''^a = \text{MultiHeadAtt}\left(Q_i'^a, h^f, h^f\right); \quad (5)$$

3) **Feed-Forward**:

$$Q_i^a = \text{FFN}\left(Q_i''^a\right), \quad (6)$$

where MultiHeadAtt(query, key, value) is the multi-head attention operation, and FFN is a postion-wise feed-forward network.

In the self-attention, we aim to learn the relationship among the labels, so the query, key and value are all from the label embeddings, while in the cross-attention, we aim to connect the labels to the fact, so the key and value become the fact embedding.

After L Transformer decoder layers, we obtain the sequence $Q_L^a \in \mathbb{R}^{l_a \times d}$, where $Q_{L,i}^a \in \mathbb{R}^d$ is

[3] We explored different threshold settings in Appendix B.

[4] For simplicity, we omit the details which are the same as in the standard Transformer decoder (Vaswani et al., 2017).

Key	Value
Article 114	*Endangering public safety by dangerous means*
	Placement of hazardous substances
	Set fire
	Explosion
Article 115	*Endangering public safety by dangerous means*
	Placement of hazardous substances
Article 118	*Damage to flammable and explosive equipment*
	Damage to electrical equipment
...	

a) A2C mapping dictionary

Key	Value
Endangering public safety by dangerous means	*Article 114*
	Article 115
Placement of hazardous substances	*Article 114*
	Article 115
Damage to flammable and explosive equipment	*Article 118*
...	

b) C2A mapping dictionary

Figure 4: The illustration of mapping dictionaries.

the representation of the input that is related to i-th law article. In other words, we inject the label embedding into the fact embedding and obtain the label-related fact representation for each label.

Through the same method, we obtain the other sequence $Q_L^c \in \mathbb{R}^{l_c \times d}$ for the charge prediction.

4.2.2 Predictors

Same as in the Sec 4.1.2, the predictors P_a and P_c take the Q_L^a and the Q_L^c as input, then calculate the predicted probability $\hat{y}_a \in \mathbb{R}^{l_a}$ and $\hat{y}_c \in \mathbb{R}^{l_c}$ for each label in each task according to its corresponding label-related fact representation.

4.2.3 Train and Inference

In the training, as there is an alignment between the two tasks, we design an extra alignment loss to better combine the tasks.

Mapping Dictionary Construction From the Code of Law [5], we can construct two mapping dictionaries called A2C and C2A, where A2C means law article to charge and C2A means charge to law article. As shown in Fig. 4, in the mapping dictionary, one law article can be mapped to one or more charges, and one charge can be mapped to one or more law articles. In judgment, the appearance of one label only means the appearance of some of its mapping labels but not all of them, so the alignment is not strict.

[5] http://www.zuiming.net/51.html

Alignment Loss Since we can not directly combine the two tasks into a single task through the mapping dictionaries, we design a novel alignment loss to utilize the mapping dictionaries. Specifically, given the predicted probability of law articles $\hat{y}_a \in \mathbb{R}^{l_a}$, we calculate the mapping probability of the i-th charge as follow:

$$\hat{y}_c^m(i) = \text{Sigmoid}(\sum_{j:i \in A2C[j]} \hat{y}_a(j)), \quad (7)$$

where $A2C[j]$ means the mapping charges of j-th law article in the A2C dictionary. In the same way, we get the mapping probability of law articles $\hat{y}_a^m$.

Then, given the two pairs $(\hat{y}_a, \hat{y}_a^m)$ and $(\hat{y}_c, \hat{y}_c^m)$ we calculate the Kullback-Leibler (KL) divergence as the alignment loss to minimize the difference between the predicted probability:

$$\mathcal{L}_{KL} = \frac{1}{2}(\mathcal{D}_{KL}(\hat{y}_a^m \| \hat{y}_a) + \mathcal{D}_{KL}(\hat{y}_c^m \| \hat{y}_c)). \quad (8)$$

The total loss is the sum of cross-entropy and the alignment loss $\mathcal{L}_{total} = \mathcal{L}_C + \mathcal{L}_{KL}$.

The inference is the same as the vanilla model.

5 Experiments

5.1 Dataset Description

Following previous works (Xu et al., 2020; Yue et al., 2021b; Feng et al., 2022b; Liu et al., 2022b), we conduct our experiments on the Chinese legal dataset CAIL2018, which is publicly available and has been widely used in the LegalAI research [6]. Each sample contains the fact description and the corresponding judgment. Unlike the previous works with single-label setting, we keep the multi-label samples that involve multiple law articles or multiple charges. The detailed statistics of the dataset are given in Tab 1. We randomly divide the dataset into training set, validation set and test set according to the ratio of 8: 1: 1.

5.2 Evaluation Metric

Accuracy of judgment prediction. To evaluate the performance of the prediction, we calculate the F1 score (Mi-F, Ma-F) and Jaccard similarity coefficient (Mi-J and Ma-J) of each task, where "Mi" refers to micro and "Ma" refers to macro. Given two sets of labels A and B, the Jaccard similarity coefficient is defined as $|A \cap B|/|A \cup B|$, where $|\cdot|$ denotes the number of elements in a set.

[6] Other Chinese legal datasets are either in a single-label setting or not accessible, and datasets from other countries is not suitable for the mapping dictionary.

Type	Result
# Sample	163,035
# Multi-Label Sample	20,347
# Law Article	121
# Charge	150
Avg. # Tokens in Fact Description	246.8
Avg. # Laws in a Multi-Label Sample	1.78
Avg. # Charges in a Multi-Label Sample	2.06

Table 1: Statistics of the dataset.

Alignment ratio. We also approximately estimate the performance of alignment. Specifically, given the label sequences a (law articles) and c (charges), a label is defined as "aligned" if any of its mapping labels exist in the other sequence. The alignment ratio (Align) is calculated as:

$$Align(a, c) = \frac{|a \cap C2A(c)| + |c \cap A2C(a)|}{|a| + |c|}. \quad (9)$$

5.3 Baselines

We consider the following methods as baselines for comparison: Firstly, we implement the vanilla model defined in Sec. 4.1 with the common encoders such as **CNN** (Kim, 2014), **Bi-LSTM** (Sutskever et al., 2014), **Transformer** (Vaswani et al., 2017) and **BERT** (Devlin et al., 2019). We also use several multi-label classification methods, where **AXML** (You et al., 2019) is a label tree-based method, **LSAN** (Xiao et al., 2019) uses a label-specific document representation, and **HTTN** (Xiao et al., 2021b) focuses on long-tailed labels. As for LJP methods, **DPAM** (Wang et al., 2018) uses pair-wise attention and a dynamic threshold to predict multiple law articles. **HMN** (Wang et al., 2019) predicts multiple law articles using the hierarchical relationship in the laws. **HAN-SGM** (Zhu et al., 2020) posits the multiple charges prediction as a sequence generation task. **FLA** (Luo et al., 2017) predicts multiple relevant law articles to support the single charge prediction.

K-LJP(CNN) and **K-LJP(BERT)** stand for the implementation of K-LJP with the encoder of CNN and BERT, respectively. We conduct ablation experiments on K-LJP(BERT) as follows: **w/o detail** means that we remove the label detail and randomly initialize the embedding of each label. **w/o align** denotes the approach such that we remove the alignment loss. **w/o d&a** means that we remove the label detail and the alignment loss. **Direct Mapping** denotes that we get the charges (law articles) by the mapping dictionary and the predicted law articles (charges), which means the two tasks are combined into one. **w cosine** indicates we replace the KL divergence with the cosine similarity. **w contrast** denotes we replace the KL divergence with the contrast loss like Yan et al. (2021).

In addition, we re-train K-LJP on the CAIL2018 dataset with single-label cases only to compare it with single-label setting LJP methods as follows[7]: **TopJudge** (Zhong et al., 2018) utilizes the topological relationship among the subtasks. **MPBFN** (Yang et al., 2019) leverages the topology structure of multiple tasks and word collocation attention for accurate prediction. **LADAN** (Xu et al., 2020) uses a graph distillation to extract discriminative features. **R-Former** (Dong and Niu, 2021) formalizes LJP as a node classification problem. **NeurJudge** (Yue et al., 2021a) splits the fact description into two parts and encodes them separately. **CEEN** (Lyu et al., 2022) propose a reinforcement learning framework for legal text mining, predicting judgment by extracting discriminative criminal elements. **CTM** (Liu et al., 2022b) enhances legal judgment prediction by using contrastive case relations. **EPM** (Feng et al., 2022b) leverages legal event information and cross-task consistency constraints for LJP.

5.4 Experimental Settings

Our experiment is carried out on two V100 GPUs, and all the baseline models adopt the settings in their original papers. We rerun the experiments five times with different random seeds and report the average. We also use the Fisher randomization test to ensure the significance of the results. Note that the set of Multi-Label Sample refers to the multi-label samples in the test set. We further explore the impact of different thresholds on task performance and the performance of LLMs on our task in Appendix.

5.5 Experiment Results

Results of judgment prediction: From Tab. 2 and Tab. 4, we can conclude that: 1) K-LJP(CNN) and K-LJP(BERT) achieve a notable improvement compared to CNN and BERT, which indicates the benefit of the injection of legal knowledge. 2) Compared to the baselines, K-LJP(BERT) achieves a better performance on both the set of All Sample and the set of Multi-Label Sample, especially on the Ma-F and Ma-J. 3) With the simple CNN model, K-LJP(CNN) demonstrates a comparable

[7]The output of K-LJP here is the label with the highest logit and all these methods are trained on the same dataset.

Method	All Sample				Multi-Label Sample			
	Mi-F	Ma-F	Mi-J	Ma-J	Mi-F	Ma-F	Mi-J	Ma-J
CNN (Kim, 2014)	84.67	75.93	78.05	65.66	82.82	49.95	70.68	41.28
Bi-LSTM (Sutskever et al., 2014)	85.43	76.68	77.66	65.99	83.07	51.33	71.04	41.99
Transformer (Vaswani et al., 2017)	85.82	75.14	75.16	64.17	81.26	49.53	68.44	40.01
BERT (Devlin et al., 2019)	89.12	81.57	83.69	74.17	86.59	55.07	76.35	46.59
AXML (You et al., 2019)	88.04	78.11	78.64	67.67	86.12	57.33	75.62	48.78
LSAN (Xiao et al., 2019)	89.13	79.92	80.39	70.22	85.04	55.39	73.98	47.11
HTTN (Xiao et al., 2021b)	88.19	77.71	78.88	67.41	84.30	53.27	72.86	44.53
FLA (Luo et al., 2017)	85.64	70.49	74.88	59.96	79.29	41.31	65.69	36.43
DPAM (Wang et al., 2018)	83.66	69.17	67.59	57.97	81.61	49.40	68.94	41.03
HMN (Wang et al., 2019)	89.29	78.57	80.65	68.62	83.61	51.06	71.83	43.13
HAN-SGM (Zhu et al., 2020)	84.30	72.03	72.86	60.54	81.28	48.47	68.46	38.88
K-LJP(CNN)	89.48	80.74	80.96	71.04	86.13	57.66	75.64	48.89
K-LJP(BERT)	**91.52****	**85.22****	**84.36***	**77.14****	**87.37***	**61.41****	**77.57***	**52.91****
w/o detail	91.34	84.07	84.06	75.74	87.06	59.25	77.08	50.63
w/o align	91.49	83.39	84.31	75.41	86.95	56.71	76.91	48.13
w/o d&a	90.75	83.09	83.74	74.66	86.61	56.34	76.38	48.54
Direct Mapping	89.48	83.84	81.43	75.43	87.21	59.24	77.33	50.85

Table 2: Results of law article prediction, the best is **bolded** and the second best is underlined. */** denotes that KLJP performs significantly better than the second-best baselines at $p < 0.05/0.01$ level.

Method	Law Article				Charge			
	Mi-F	Ma-F	Mi-J	Ma-J	Mi-F	Ma-F	Mi-J	Ma-J
K-LJP	**87.37**	**61.41**	**77.57**	**52.91**	**88.04**	58.68	**78.63**	50.60
w cosine	86.98	56.85	76.97	48.47	87.88	**59.05**	78.38	**51.11**
w contrast	85.81	55.89	75.14	46.77	86.51	54.01	76.22	44.96

Table 3: Results of different alignment losses.

result across all metrics, especially on the set of Multi-Label Sample. 4) The prediction accuracy of the law article is a little higher than the charge; this may be because there are more charges than law articles. 5) Compared to the set of All Sample, in the set of Multi-Label Sample, the Ma-F score drops in both tasks (e.g., the Ma-F of LSAN drops 24.53%, and the Mi-F drops only 4.09%), which shows the difficulty of the multi-label classification.

Tab. 3 demonstrates the results of different alignment losses on the set of Multi-Label Sample. With the KL divergence, K-LJP achieves a better result on most of the metrics, and cosine loss is better than contrast loss in this task.

Based on Tab. 2 and Tab. 4 in the ablation experiments, we make the following observations: 1) The performance gap between K-LJP (BERT) and the w/o detail indicates the effectiveness of label detail (e.g., Ma-F drops from 61.41% to 59.25% in the set of Multi-Label Sample in the prediction of law articles), and the gap between K-LJP (BERT) and the w/o align shows the effectiveness of the proposed alignment loss. 2) Compared to BERT, w/o d&a illustrates the importance of learning the relationships among the labels. 3) The prediction of law article benefits more from the alignment (e.g., Ma-F drops from 61.41% to 56.71% in the set of Multi-Label Sample in the prediction of law article, which decreases only 0.9% in the prediction

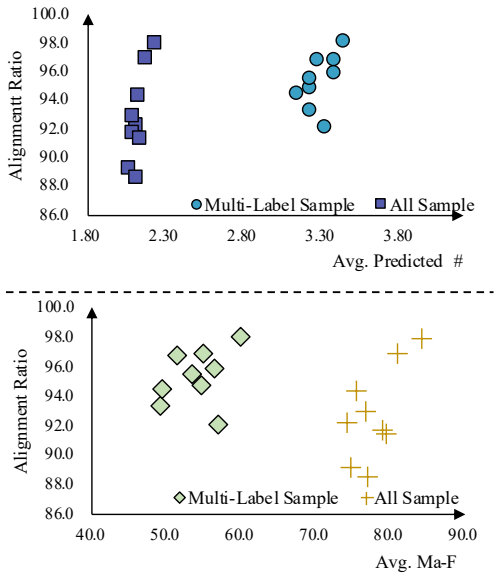

Figure 5: a) The relevance of the predicted label number and the alignment ratio. b) The relevance of Ma-F and the alignment ratio, we average the Ma-F of two tasks.

of charge). 4) Direct Mapping suggests that combining two tasks into one can hurt performance.

Results in single-label setting: From Tab. 5, we can conclude that: 1) Compared to CNN and BERT, the performance improvement of K-LJP is not as high as it is in the multi-label setting, which may be because the alignment is less challenging in the single-label setting. 2) Compared to other SOTA LJP methods, K-LJP(BERT) still achieves

Method	All Sample				Multi-Label Sample			
	Mi-F	Ma-F	Mi-J	Ma-J	Mi-F	Ma-F	Mi-J	Ma-J
CNN (LeCun et al., 1989)	85.71	73.61	75.54	63.24	82.96	48.60	71.35	40.51
Bi-LSTM (Sutskever et al., 2014)	85.55	74.56	75.29	63.84	83.47	51.78	72.11	42.53
Transformer (Vaswani et al., 2017)	84.78	73.54	73.58	62.44	82.47	49.24	70.16	40.27
BERT (Devlin et al., 2019)	89.43	80.53	82.54	73.12	87.09	55.04	77.13	46.49
AXML (You et al., 2019)	87.16	76.13	77.25	65.78	87.64	56.82	76.59	48.35
LSAN (Xiao et al., 2019)	88.15	78.38	78.81	68.54	85.71	54.20	74.99	46.17
HTTN (Xiao et al., 2021b)	87.17	76.08	77.25	65.76	85.49	53.50	74.66	45.10
HAN-SGM (Zhu et al., 2020)	83.27	70.71	71.33	59.18	82.00	46.52	69.49	37.55
K-LJP(CNN)	88.46	78.53	79.31	68.58	86.41	55.26	76.07	46.30
K-LJP(BERT)	**90.84***	**83.39****	**83.22***	**75.21****	**88.04**	**58.68****	**78.63****	**50.60****
w/o detail	90.67	82.26	82.93	73.67	88.01	57.94	78.58	49.90
w/o align	90.78	82.42	83.12	74.18	87.75	57.78	78.18	49.59
w/o d&a	90.42	81.67	82.52	73.18	87.15	57.35	77.23	48.95
Direct Mapping	83.25	81.80	79.50	71.69	81.32	55.33	68.52	46.17

Table 4: Results of charge prediction, the best is **bolded** and the second best is underlined. */** denotes that KLJP performs significantly better than the second-best baselines at $p < 0.05/0.01$ level.

Method	Law Article		Charge	
	Mi-F	Ma-F	Mi-F	Ma-F
CNN (2014)	89.30	80.17	88.15	80.51
BERT (2019)	90.21	82.16	89.67	81.19
TopJudge (2018)	86.85	78.68	88.42	80.41
MPBFN (2019)	86.30	78.84	87.69	81.59
LADAN (2020)	88.92	80.46	89.45	81.19
R-former (2021)	**90.59**	83.62	90.43	83.21
NeurJudge (2021)	89.94	82.26	90.31	81.59
CEEN (2022)	89.85	80.80	90.45	81.86
EPM (2022)	89.87	81.43	90.17	81.22
CTM (2022)	88.50	79.85	**90.76**	82.35
K-LJP(CNN)	89.24	81.35	90.18	80.66
K-LJP(BERT)	90.22	**84.28**	90.46	**83.27**

Table 5: Results of single-label setting LJP methods. We modify K-LJP to the single-label setting here.

Method	All Sample		Multi-Label Sample	
	Lab. #	Align	Lab. #	Align
CNN	2.07	89.19	3.22	93.33
Bi-LSTM	2.13	94.38	3.38	96.79
Transformer	2.11	92.21	3.14	94.51
BERT	2.17	96.93	3.27	96.85
AXML	2.11	88.53	3.32	92.05
LSAN	2.09	91.73	3.23	94.78
HTTN	2.09	92.92	3.22	95.47
K-LJP(CNN)	2.14	91.39	3.38	95.90
K-LJP(BERT)	2.23	**97.93**	3.44	**98.08**

Table 6: Results of Alignment Ratio. Lab. # refers to the average number of predicted labels.

competitive performance in the single-label setting.

Results of alignment ratio: 1) From Tab. 6, we find that K-LJP(BERT) achieves the best alignment ratio, especially on the multi-label sample set. 2) Looking at Fig. 5 (a), we observe that the average number of predicted label lengths of different methods tends to be approximate. In other words, the alignment ratio is not affected by the average number of predicted labels. 3) From Fig. 5 (b), we find that high accuracy (e.g., Ma-F) does not equal to a high alignment ratio; this may result from the ignorance of the connection between the two tasks.

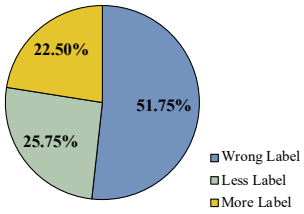

Figure 6: The distribution of the three error types.

5.6 Error Study

We also conduct an error analysis to explore the limitations of our technique. We randomly select 200 samples with wrong results, then we manually determine the error types and the reasons behind the errors. From Fig. 6, we find that the most frequent error type is the "wrong label" (51.75%), which means the model predicts the correct label number, but the wrong labels. The other two error types (e.g., "less label" and "more label") make up the remaining half. We conclude the reason behind the error as follows: 1) Labels may have similar definitions, when it comes to an unbalanced distribution, the model tends to output the label with high frequency rather than the infrequent label (e.g., Robbery vs. Plunder). 2) The occurrence of certain words will mislead the model. For example, when the word "ID card" appears, the model has a high preference to the charge of "Forgery of fake certificates", even if there is no corresponding fact.

To address these problems, a promising way is to summarize the input at first and then use a debiasing method to distinguish the confusing labels.

6 Conclusion and Future Work

In this paper, we investigate the prediction of law articles and charges in the legal judgment prediction

(LJP) task posited as a multi-task and multi-label problem. By incorporating legal knowledge, we propose a novel K-LJP approach. Specifically, we inject "label-level knowledge" into the encoder by utilizing the label definitions and learning the complex relationships among the labels and between the label and the case's fact. Then, since an alignment exists between law articles and charges, we design an alignment loss to integrate "task-level knowledge" into the model. Experiments on a real-world dataset show the effectiveness of K-LJP.

In the future, we will explore the following directions: 1) using similar cases to help the prediction. 2) predicting the term of penalty based on the predicted results of the law articles and charges.

7 Ethical Statement

With the development of AI, more and more LegalAI technologies are being proposed to assist judges, especially those who suffer from an intense workload (Lin et al., 2012; Zhong et al., 2018; Chalkidis et al., 2019). LegalAI is a vital but sensitive area, hence any subtle miscalculation may trigger serious consequences, so it is imperative to discuss the related ethical issues. Our model aims to predict the judgment based on the case fact, which is an algorithmic investigation where still many potential risks remain (e.g., demographic bias, lack of interpretability). The algorithm will be beneficial to achieve the goal of "treating like cases alike" (Sun et al., 2020). Nevertheless, the algorithm only intends to assist judges and should never "replace" human judges to deliver sentences; the judges must always conduct manual verification and judgment of the results predicted by any algorithm.

8 Limitations

In this section, we discuss the limitations of our work which are as follows:

- The model is based on legal knowledge, hence once the Code of Law is revised, adjusting correspondingly the knowledge of the model remains a challenge.
- The performance of the models can be affected by the label imbalance, especially for low-frequency labels.

9 Acknowledgments

This work was supported by the National Natural Science Foundation of China (62376243, 62441605, 62037001), and the Starry Night Science Fund at Shanghai Institute for Advanced Study (Zhejiang University).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in english. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4317–4323. Association for Computational Linguistics.

Junyun Cui, Xiaoyu Shen, Feiping Nie, Zheng Wang, Jinglong Wang, and Yulong Chen. 2022. A survey on legal judgment prediction: Datasets, metrics, models and challenges. *CoRR*, abs/2204.04859.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Qian Dong and Shuzi Niu. 2021. Legal judgment prediction via relational learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 983–992.

Xinyu Duan, Yating Zhang, Lin Yuan, Xin Zhou, Xiaozhong Liu, Tianyi Wang, Ruocheng Wang, Qiong Zhang, Changlong Sun, and Fei Wu. 2019. Legal summarization for multi-role debate dialogue via controversy focus mining and multi-task learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1361–1370. ACM.

Yi Feng, Chuanyi Li, and Vincent Ng. 2022a. Legal judgment prediction: A survey of the state of the art. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5461–5469.

Yi Feng, Chuanyi Li, and Vincent Ng. 2022b. Legal judgment prediction via event extraction with constraints. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 648–664, Dublin, Ireland. Association for Computational Linguistics.

Leilei Gan, Kun Kuang, Yi Yang, and Fei Wu. 2021. Judgment prediction via injecting legal knowledge into neural networks. In *AAAI Conference on Artificial Intelligence*.

R Keown. 1980. Mathematical models for legal prediction. *Computer/lj*, 2:829.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551.

Ang Li, Qiangchao Chen, Yiquan Wu, Ming Cai, Xiang Zhou, Fei Wu, and Kun Kuang. 2024a. From graph to word bag: Introducing domain knowledge to confusing charge prediction.

Ang Li, Yiquan Wu, Yifei Liu, Fei Wu, Ming Cai, and Kun Kuang. 2024b. Enhancing court view generation with knowledge injection and guidance.

Wan-Chen Lin, Tsung-Ting Kuo, Tung-Jia Chang, Chueh-An Yen, Chao-Ju Chen, and Shou-de Lin. 2012. Exploiting machine learning models for chinese legal documents labeling, case classification, and sentencing prediction. *Int. J. Comput. Linguistics Chin. Lang. Process.*, 17(4).

Dugang Liu, Weihao Du, Lei Li, Weike Pan, and Zhong Ming. 2022a. Augmenting legal judgment prediction with contrastive case relations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2658–2667.

Dugang Liu, Weihao Du, Lei Li, Weike Pan, and Zhong Ming. 2022b. Augmenting legal judgment prediction with contrastive case relations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2658–2667, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. 2021. Query2label: A simple transformer way to multi-label classification. *CoRR*, abs/2107.10834.

Yifei Liu, Yiquan Wu, Ang Li, Yating Zhang, Changlong Sun, Weiming Lu, Fei Wu, and Kun Kuang. 2024. Unleashing the power of LLMs in court view generation by stimulating internal knowledge and incorporating external knowledge. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2782–2792, Mexico City, Mexico. Association for Computational Linguistics.

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 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2727–2736. Association for Computational Linguistics.

Yougang Lyu, Zihan Wang, Zhaochun Ren, Pengjie Ren, Zhumin Chen, Xiaozhong Liu, Yujun Li, Hongsong Li, and Hongye Song. 2022. Improving legal judgment prediction through reinforced criminal element extraction. *Information Processing & Management*, 59(1):102780.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale multi-label text classification - revisiting neural networks. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II*, volume 8725 of *Lecture Notes in Computer Science*, pages 437–452. Springer.

Shounak Paul, Pawan Goyal, and Saptarshi Ghosh. 2021. Lesicin: A heterogeneous graph-based approach for automatic legal statute identification from indian legal documents.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018, page 3132. NIH Public Access.

Changlong Sun, Yating Zhang, Qiong Zhang, and Xiaozhong Liu. 2020. Legal artificial intelligence - have you lost a piece from jigsaw puzzle? In *Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice, AAAI-MAKE 2020, Palo Alto, CA, USA, March 23-25, 2020, Volume I*, volume 2600 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Suxin Tong, Jingling Yuan, Peiliang Zhang, and Lin Li. 2024. Legal judgment prediction via graph boosting with constraints. *Information Processing Management*, 61(3):103663.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 325–334. ACM.

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 *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 485–494. ACM.

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 2020, Online, November 16-20, 2020*, pages 763–780. Association for Computational Linguistics.

Yiquan Wu, Yifei Liu, Weiming Lu, Yating Zhang, Jun Feng, Changlong Sun, Fei Wu, and Kun Kuang. 2022. Towards interactivity and interpretability: A rationale-based legal judgment prediction framework. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4787–4799.

Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021a. Lawformer: A pre-trained language model for chinese legal long documents. *AI Open*, 2:79–84.

Guangyi Xiao, Even Chow, Hao Chen, Jiqian Mo, Jingzhi Guo, and Zhiguo Gong. 2017. Chinese questions classification in the law domain. In *14th IEEE International Conference on e-Business Engineering, ICEBE 2017, Shanghai, China, November 4-6, 2017*, pages 214–219. IEEE Computer Society.

Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-specific document representation for multi-label text classification. 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, Hong Kong, China, November 3-7, 2019*, pages 466–475. Association for Computational Linguistics.

Lin Xiao, Xiangliang Zhang, Liping Jing, Chi Huang, and Mingyang Song. 2021b. Does head label help for long-tailed multi-label text classification. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14103–14111. AAAI Press.

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 2020, Online, July 5-10, 2020*, pages 3086–3095. Association for Computational Linguistics.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5065–5075. Association for Computational Linguistics.

Wenmian Yang, Weijia Jia, Xiaojie Zhou, and Yutao Luo. 2019. Legal judgment prediction via multi-perspective bi-feedback network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4085–4091. ijcai.org.

Feng Yao, Chaojun Xiao, Xiaozhi Wang, Zhiyuan Liu, Lei Hou, Cunchao Tu, Juanzi Li, Yun Liu, Weixing Shen, and Maosong Sun. 2022. LEVEN: A large-scale chinese legal event detection dataset. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 183–201. Association for Computational Linguistics.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5812–5822.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021a. Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 973–982. ACM.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021b. 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*, pages 973–982.

Qihui Zhao, Tianhan Gao, Song Zhou, Dapeng Li, and Yingyou Wen. 2022. Legal judgment prediction via heterogeneous graphs and knowledge of law articles. *Applied Sciences*, 12(5):2531.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 3540–3549. Association for Computational Linguistics.

Xiang Zhou, Yudong Wu, Ang Li, Ming Cai, Yiquan Wu, and Kun Kuang. 2024. Unlocking authentic judicial reasoning: A template-based legal information generation framework for judicial views. *Knowledge-Based Systems*, 301:112232.

Kongfan Zhu, Baosen Ma, Tianhuan Huang, Zeqiang Li, Haoyang Ma, and Yujun Li. 2020. Sequence generation network based on hierarchical attention for multi-charge prediction. *IEEE Access*, 8:109315–109324.

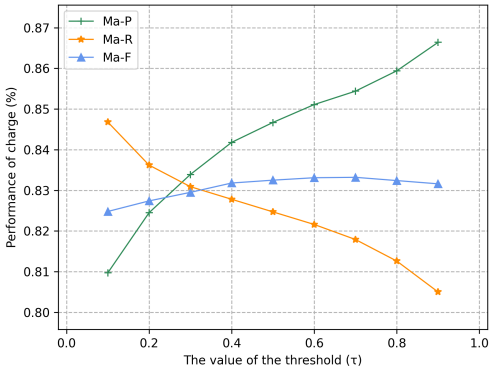

Figure 7: The impact of different thresholds on the performance of charge prediction.

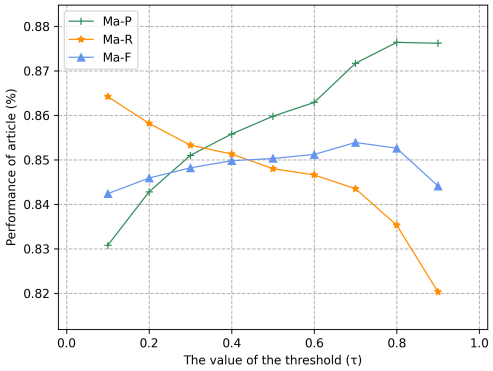

Figure 8: The impact of different thresholds on the performance of law article prediction.

A The Impact of Different Thresholds

We describe here the experiments conducted with different thresholds of our K-LJP(BERT) method. From Fig. 7 and Fig. 8, it can be observed that: (1) As the threshold gradually increases, Ma-P (Precision) increases while Ma-R (Recall) decreases. This is because a higher threshold leads to fewer predicted labels, making it easier to match the correct labels. Conversely, a lower threshold results in more predicted labels, increasing the likelihood of covering all true labels. (2) The highest Ma-F (F1-score) is achieved when the threshold is set to 0.7. However, at this threshold, there is a significant difference between Ma-P and Ma-R, which can compromise the model's effectiveness. A threshold of 0.5 yields a considerable Ma-F score, with a smaller difference between Ma-P and Ma-R. Therefore, we adopt a threshold of 0.5 for our method and for all other baselines.

B The performance of LLMs

Methods	Mi-F	Ma-F	Mi-J	Ma-J
GPT-4	0.62	0.13	0.31	0.17
Claude-3	3.95	2.30	1.61	0.93
KLJP(BERT)	91.52	85.22	84.36	77.14

Table 7: Performance of LLMs in Law article prediction

Methods	Mi-F	Ma-F	Mi-J	Ma-J
GPT-4	33.78	20.16	20.32	16.12
Claude-3	34.05	21.85	21.28	16.18
KLJP(BERT)	90.84	83.39	83.22	75.21

Table 8: Performance of LLMs in charge prediction

We conducted experiments comparing the performance of LLMs (GPT-4 and Claude-3) and our K-LJP (BERT) in legal judgment prediction (LJP). We performed experiments on the entire test set. We concatenated factual descriptions with carefully designed prompts, and then input them into the LLM to generate responses. For the charge prediction task, the prompt was: *Based on the above criminal facts, please predict the defendant's criminal charge and return a list, such as ['Theft', 'Intentional Injury']. Do not provide any additional content.* For the law article prediction task, the prompt was: *Based on the above criminal facts, please predict the applicable criminal law article and return a list, such as ['Article 273', 'Article 264']. Do not provide any additional content.*

From the results shown in Tab. 7 and Fig. 8, for both law article prediction task and charge prediction task, it can be observed that: (1) K-LJP (BERT) significantly outperforms LLMs across all metrics. In law article prediction, K-LJP achieves a Mi-F score of 91.52, far exceeding GPT-4's 0.62 and Claude-3's 3.95. Similarly, for charge prediction, K-LJP scores 90.84, while GPT-4 and Claude-3 reach only 33.78 and 34.05, respectively. (2) These results highlight the limitations of LLMs in tasks involving labels with less substantive meaning, such as law article indices, whereas K-LJP, designed for legal data, excels by effectively handling the many-to-many relationships between charges and legal articles. LLMs operate within a generation paradigm, where they must select labels from a large vocabulary during prediction tasks, leading to poor performance on prediction tasks. Specifically, the law article labels are numerical and lack semantic meaning, while the charge labels carry inherent

Fact Description	Ground Truth	NeurJudge	K-LJP (BERT)
The defendant, W, drove a tricycle to a foot bath shop in Cixi City and **stole** a white Apple 4S phone worth RMB 750 while the victim, L, was asleep. As W was **escaping**, L woke up and grabbed the tricycle. Despite knowing this, W accelerated, **dragging** L for about 20 meters and causing L to fall and **sustain minor injuries**. The **stolen** phone was recovered and returned to L.	Article 263, Article 269, Robbery	Article 264, Robbery	Article 263, Robbery

Table 9: Legal Case Analysis Table

Method	Parameters	Training	Inference
LADAN	29.7 M	1505s	43s
R-former	47.6 M	1692s	52s
K-LJP (Bert)	32.4 M	1269s	39s

Table 10: Resource Consumption

semantic significance. As a result, LLMs perform worse on law article prediction.

C Detailed error analysis on case

In this case, the defendant was found stolen but exhibited violent behavior ("dragging"), constituting "Transformed Robbery" and offending "Article 263". K-LJP correctly predicted the charge and one law article, while NeurJudge, although accurate in predicting the charge, mistakenly predicted "Article 264", which applies to "Theft", due to the disruption caused by terms like "stole" and "escape". This reflects the importance of task-level alignment. However, how to guide the model to focus on transformed law articles such as "Article 269" remains an area for further exploration.

D Resource Consumption

Our method uses two parallel transformer decoders, each with 4 layers, resulting in a compact model and low time consumption. We compared the model size and time consumption of our method with the baselines using 2 V100 GPUs. Following the complexity calculation method of GJudge (Tong et al., 2024), our method (K-LJP (BERT)) includes an encoder, transformer decoder, and predictor, with a total complexity of $O\left(|n^2|\right)$. The complexity of self-attention, feed-forward, cross-attention, predictor are $O\left(|n^2d + nd^2|\right)$, $O\left(|nd^2|\right)$, $O\left(|n^2d|\right)$, $O\left(|nd|\right)$ respectively.