Case Retrieval for Legal Judgment Prediction in Legal Artificial Intelligence

Han Zhang
School of Information,
Renmin University of China.
zhanghanjl@ruc.edu.cn

Zhicheng Dou
Gaoling School of Artificial Intelligence,
Renmin University of China.
dou@ruc.edu.cn

Abstract

Legal judgment prediction (LJP) is a basic task in legal artificial intelligence. It consists of three subtasks, which are relevant law article prediction, charge prediction and term of penalty prediction, and gives the judgment results to assist the work of judges. In recent years, many deep learning methods have emerged to improve the performance of the legal judgment prediction task. The previous methods mainly improve the performance by integrating law articles and the fact description of a legal case. However, they rarely consider that the judges usually look up historical cases before making a judgment in the actual scenario. To simulate this scenario, we propose a historical case retrieval framework for the legal judgment prediction task. Specifically, we select some historical cases which include all categories from the training dataset. Then, we retrieve the most similar Top-k historical cases of the current legal case and use the vector representation of these Top-k historical cases to help predict the judgment results. On two real-world legal datasets, our model achieves better results than several state-of-the-art baseline models.

1 Introduction

With the rapid development of artificial intelligence, it has become a trend to use artificial intelligence to help judicial personnel. Legal judgment prediction (LJP) is such an artificial intelligence task in legal artificial intelligence. As shown in Table 1, given the fact description of a legal case, the legal judgment prediction task can provide the judgment result of the case. The predicted result consists of three parts: relevant law article, charge and term of penalty. Legal judgment prediction can not only give the judgment results efficiently for reference for judicial personnel but also provide legal suggestions for ordinary people when there is a legal dispute (Zhong et al., 2020; Wang et al., 2019; Zhong et al., 2018; Zhang et al., 2021) in daily life.

With the application of deep learning in legal artificial intelligence, various methods have been proposed to improve the performance of legal judgment prediction. Some methods (Zhong et al., 2018; Yang et al., 2019) consider using the order information among the three subtasks of legal judgment prediction in reality to improve the representation of fact description. Further, some methods (Yue et al., 2021; Ma et al., 2021; Feng et al., 2022) consider a fine-grained division of the fact description to improve the fact representation. Additionally, some methods (Luo et al., 2017; Hu et al., 2018; Wang et al., 2019; Xu et al., 2020) consider the important role of law articles in reality and introduce them to improve the performance. These efforts have effectively improved the performance of legal judgment prediction. However, the existing methods are affected by the fact that the law articles are too concise and still have limitations in modelling the judgment process.

On the one hand, the law articles are very concise and lack specific details and some law articles have similar provisions and so easy to be confused. As shown in Figure 1, *Article #114* and *Article #115* both stipulate the same charge *Crime of Arson* and the provisions in the two law articles are very short. In order to distinguish them, the judge usually re-finds and analyzes historical cases because the fact description information of historical cases usually contains more detailed information than the law articles.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Fact Description On XX, XXX, the procuratorate accused the defendant Yang XX of taking gasoline out of his motorcycle fuel tank after quarrelling with his girlfriend Tang XX, putting it into a beer bottle, and **pouring gasoline through the crack of the door** into room X of the rental room opposite the XX Internet cafe in the XX community where Tang XX is located, and **using a lighter to ignite the gasoline**. **The fire spread to the room along with the gasoline and was extinguished** by Tang XX and other people in the room. On the morning of that day, the public security police arrested the defendant Yang XX ...
Relevant Law Article *Article #114* **[Crime of Arson] Whoever commits arson**, breaches a dike, causes an explosion, spreads toxic, radioactive, infectious disease pathogens and other substances or endangers public security by other dangerous methods, but has not caused serious consequences, shall be sentenced to **fixed-term imprisonment of not less than three years but not more than ten years.**
Charge: **Crime of Arson**
Term of Penalty: **A fixed-term imprisonment of thirty-six months**

Table 1: An example of the legal judgment prediction task.

On the other hand, most of the previous methods predict the judgment results mainly based on the fact description of a single case, however, they overlook the practical scenario that judges usually look up typical historical cases for reference before making a judgment. As we all know, historical cases are very important for making a judgment, whether in the Case Law system or the Statutory Law system.[0] In the Case Law system, judges mainly refer to historical cases to make a judgment. In the Statutory Law system, before making a judgment the judges should not only look up the law articles but also look up typical historical cases. Obviously, historical cases are indispensable references for judges in their work.

To solve these challenges, we propose a framework for legal judgment prediction based on a historical case retrieval module to simulate the actual legal scenario of looking up historical cases before making a judgment.

First, we consider that the number of cases looked up by judges in actual work is usually limited and select a part of cases from the training dataset as historical cases.

Second, in order to avoid the impact of highly unbalanced class distribution of the dataset on the model performance (Hu et al., 2018; Zhang et al., 2021), we consider selecting the same number of historical cases for each category.

Third, we retrieve the most similar Top-k historical cases of the current legal case and concatenate the vector representation of these Top-k cases and the fact description of the current case to predict the judgment results. Finally, we train our model with a cross-entropy loss function. We call our model **CR4LJP**, which stands for **C**ase **R**etrieval framework for **L**egal **J**udgment **P**rediction.

Our contributions are three-fold:

(1) We take into account that judges usually look up historical cases before making a judgment after investigating the human justice system.

(2) We propose a case retrieval framework for the legal judgment prediction task to use historical cases to help predict the judgment results.

(3) Experiment results of our framework with different encoders on two real large-scale legal datasets are better than the state-of-the-art models and verify the effectiveness of our framework. This study shows that case retrieval is an effective way to improve the performance of the legal judgment prediction task.

2 Related Work

2.1 Legal Judgment Prediction

The earliest legal judgment prediction (LJP) methods (Kort, 1957; Ulmer, 1963; Nagel, 1963; Segal, 1984; Gardner, 1984) mainly use mathematical and statistical tools. These methods are based on artificial features or rules, so they are difficult to extend. In recent years, some researchers have proposed a lot of models (Zhong et al., 2020; Zhong et al., 2018; Yang et al., 2019; Dong and Niu, 2021) based on

[0]The details of the Case Law and Statutory Law system can be found in https://en.wikipedia.org/wiki/Case_law and https://en.wikipedia.org/wiki/Statutory_law.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.

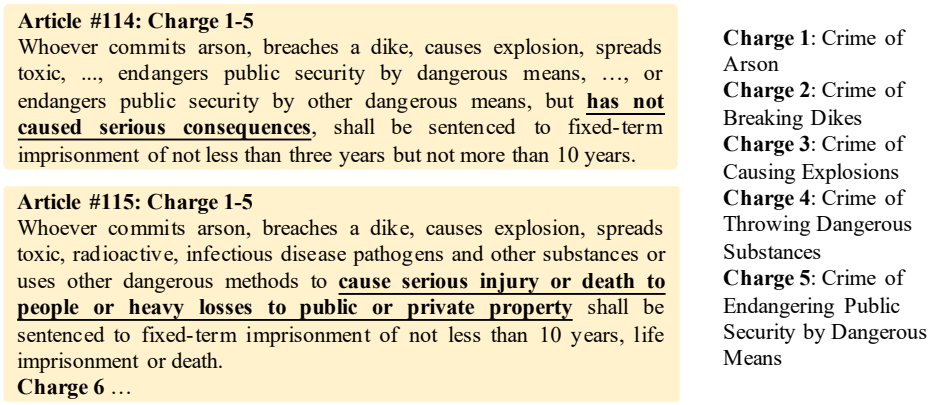

Figure 1: Article #114 and Article #115 both stipulate the same charges (Charge 1-5). There is little difference between the specific provisions of Article #114 and Article #115 on the Crime of Arson.

deep learning to predict judgment results. Specifically, some research works (Zhong et al., 2018; Yang et al., 2019) consider that the legal judgment prediction task is composed of three subtasks, and there are dependencies among them which are useful information. Some research works (Yue et al., 2021; Ma et al., 2021; Feng et al., 2022) consider that the fact description is usually long, and the fact description can be better represented by dividing or extracting the fine-grained information. Some research works (Luo et al., 2017; Hu et al., 2018; Wang et al., 2019; Xu et al., 2020) consider the important role of law articles in reality and then study how to make use of the information of law articles. These works improve the performance of LJP, but they fail to take into account that historical cases are also important information.

2.2 Retrieval Methods

For deep learning models, even the pre-trained models, such as Bert (Devlin et al., 2019), can not remember all samples. Therefore, it is worth considering using a retrieval model to obtain additional information. Generally, retrieval models can be divided into two types: sparse representation based on bag-of-word (BOW) (Chen et al., 2017) and dense vector representation based on neural networks (Karpukhin et al., 2020; Zhou et al., 2020). The retrieval models based on sparse representation have been applied in machine translation (Gu et al., 2018) and open domain question answering (Chen et al., 2017; Wang et al., 2018; Lin et al., 2018). The retrieval models based on dense vector representation (Karpukhin et al., 2020; Zhou et al., 2020) have received more attention in recent years. This method can achieve better recall performance than the sparse retrieval model on various Natural Language Processing (NLP) tasks, such as personalized search (Ma et al., 2020; Zhou et al., 2020) and domain question answering (Karpukhin et al., 2020; Guu et al., 2020; Yu et al., 2022). Considering that judges usually only need some typical cases and the good performance of dense vector representation, we use the dense vector representation retrieval method.

3 Problem Definition

Before introducing our model, we first introduce some concepts and definitions of legal judgment prediction.

A **legal case** in our paper consists of a fact description and three judgment results, which are made by human judges. The **fact description** is a text that describes the criminal facts of a suspect. As shown in Figure 2, our model uses f to represent it. The three **judgment results** are relevant law article, charge and term of penalty and we use y_1, y_2 and y_3 to represent them respectively. Then a legal case can be represented as:

$$\text{Case} = (f, y_1, y_2, y_3), \tag{1}$$

where f, y_1, y_2, y_3 are defined above.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.

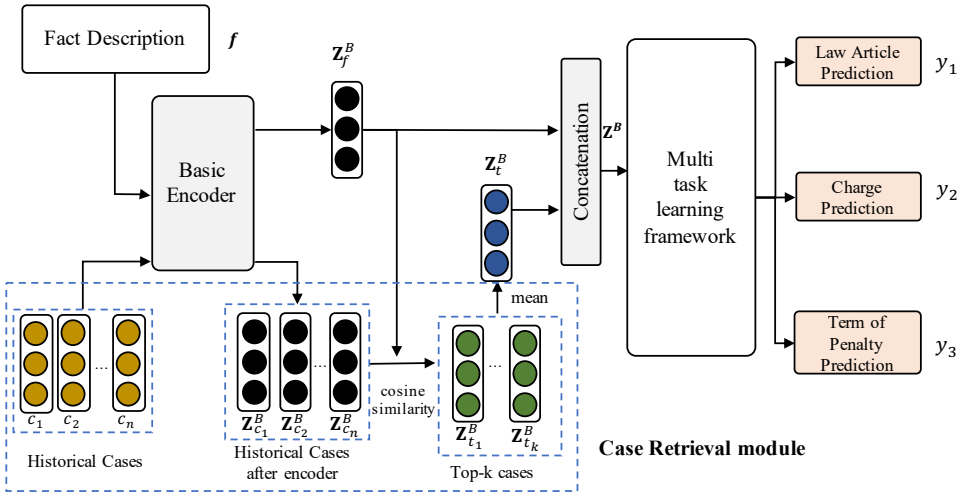

Figure 2: The framework of our model. The main module of our framework is the Basic Encoder and the Case Retrieval module.

Referring to previous studies (Zhong et al., 2018; Xu et al., 2020; Luo et al., 2017), we adopt a multi-task learning framework to solve the legal judgment prediction task. Our goal is to train a model $F(\cdot)$ which can be used to predict a case f_t in the test dataset with a given training dataset D, namely:

$$F(f_t) = (\hat{y}_1, \hat{y}_2, \hat{y}_3), \tag{2}$$

where $\hat{y}_1$, $\hat{y}_2$ and $\hat{y}_3$ are the predicted judgment results. Consistent with the existing works (Zhong et al., 2018; Xu et al., 2020), we only consider the legal cases with one relevant law article and one charge label.

4 Model Framework

In the actual judgment process, judges usually look up some typical historical cases for reference. To simulate this process, we propose a framework (CR4LJP) with a historical case retrieval module.

4.1 Overview

Our model framework is shown in Figure 2. In general, our model is a multi-task learning framework, which jointly solves three legal judgment prediction subtasks, with the case retrieval module we proposed. The main modules and training process of our model framework are as follows:

(1) The fact description f is converted into vector representation Z_f^B through the basic encoder.

(2) All selected historical cases are transformed into vector representations by the basic encoder. We select the Top-k cases which are most similar to the vector Z_f^B from these cases according to the cosine similarity. Then, we get the mean vector Z_t^B of these Top-k cases as auxiliary information.

(3) The representation vector Z_f^B and the mean vector Z_t^B of these Top-k cases are concatenated to solve the three legal judgment prediction subtasks.

(4) Our model is optimized by the losses of three subtasks. In the test phase, we also use the historical case vectors as auxiliary information to predict the judgment results.

4.2 Basic Encoder

As shown in Figure 2, our model framework uses the same encoder for the current case and historical cases. Considering the consistency with the previous models (Xu et al., 2020; Zhong et al., 2018; Yue et al., 2021) and the operation efficiency, we adopt the recurrent neural network (RNN) based encoder. Although we use RNN based encoder, our framework can flexibly select the neural network. Other neural networks, such as the current neural network (CNN) and pre-trained language models (PLMs), can also be used as encoders.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.

Specifically, the fact description of a legal case with m words is represented as:

$$f = (w_1, \cdots, w_m), \tag{3}$$

where w_i is a word in the fact description. Then we convert it to a word embedding sequence $\mathbf{f}$ though looking up a pre-trained word embedding table $\mathbf{E}$:

$$\mathbf{f} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_m], \mathbf{e}_i \in \mathbf{E}, \tag{4}$$

where $\mathbf{f} \in \mathbf{R}^{m \times d_e}$, and $\mathbf{e}_i \in \mathbf{R}^{d_e}$ is the embedding vector of the i-th word w_i. Then we use Bi-GRU neural network to encode the fact description.

$$\mathbf{Z_f^B} = \text{Bi-GRU}(\mathbf{f}), \tag{5}$$

where $\mathbf{Z_f^B} = (h_1, \cdots, h_l) \in \mathbf{R}^{l \times d_h}$, d_h is the length of the hidden layer of Bi-GRU encoder.

After introducing RNN based encoder, we introduce an alternative neural network Bert (Devlin et al., 2019) as the encoder. First, the fact description $\mathbf{f}$ is set as the input of Bert after an embedding layer. After the multi-layer self-attention encoder, the output of "[CLS]" token of Bert is set as the vector representation of the fact description. It can also be represented as:

$$\mathbf{Z}_f^{\text{Bert}} = \text{BERT}(f)_{[\text{CLS}]}, \tag{6}$$

where "[CLS]" is one of the tokens ouput by the Bert model.

4.3 Case Retrieval Module

In the actual judgment process, judges usually look up some typical historical cases as references. So we design a case retrieval module to simulate the scenario. As the performance of the dense representation retrieval method is usually better, we choose the dense representation retrieval method for our retrieval module.

Case Selection. In reality, the number of historical cases is huge and judges usually only look up some cases as references. For the efficiency of the model, we consider only selecting part of the cases instead of all the cases in the training dataset as historical cases to be retrieved. It should be noted that some law articles stipulate the same charges as shown in Figure 1. And considering the unbalanced distribution of categories, we select the same number of cases for each charge under each law article.

Case Retrieval. In order to realize the historical case retrieval module, we first represent all historical cases $(c_1, c_2, \cdots, c_n)$ as word embedding sequences through Formula 4, and then represent them as n encoded vectors $(\mathbf{Z}_{c_1}^B, \cdots, \mathbf{Z}_{c_n}^B)$ through Formula 5. Then we calculate the similarity scores of these n historical cases and the fact vector representation $\mathbf{Z}_f^B$ of the current case according to cosine similarity. Finally, we select the **Top-k** most similar cases as the reference cases by ranking the similarity scores, and then we calculate the mean vector of these Top-k cases:

$$\mathbf{Z}_t^B = \text{Mean}(\mathbf{Z}_{t_1}^B, \cdots, \mathbf{Z}_{t_k}^B). \tag{7}$$

The final mean vector $\mathbf{Z}_t^B$ of these **k** historical cases is the output of the case retrieval module.

4.4 Prediction and Optimization

Before predicting the judgment results for calculating the losses of three legal judgment prediction subtasks, we concatenate the vector representation of the current case and the historical cases as follows:

$$\mathbf{Z}^B = [\mathbf{Z}_f^B; \mathbf{Z}_t^B], \tag{8}$$

and then we use a multi-layer perceptron layer to predict the results as follows:

$$y_i = \text{MLP}_i(\mathbf{Z}^B), \tag{9}$$

Table 2: The statistics of the CAIL dataset.

Dataset	CAIL-small	CAIL-big
# Training Set Cases	106,750	1,648,600
# Test Set Cases	25,652	200,449
# Law Articles	94	115
# Charges	109	129
# Term of Penalty	11	11

where i represent the i-th subtask of legal judgment prediction.

Total loss. The legal judgment prediction task includes three subtasks (relevant law article prediction, charge prediction and term of penalty prediction). We use the cross-entropy loss to calculate the loss of each subtask and train our model. The total loss is calculated as follows:

$$\mathcal{L}_{\mathrm{LJP}} = -\sum_{i=1}^{3} \alpha_i \sum_{j=1}^{|N_j|} y_{i,j} \log(\hat{y}_{\mathrm{i,j}}), \tag{10}$$

where $|N_{ij}|$ represent the number of labels of subtask i, and α_i is the weight of subtask i which is hyper parameter.

5 Experiments

5.1 Datasets and Preprocessing

Most of the state-of-the-art methods for legal judgment prediction are tested on the Chinese AI and Law challenge (CAIL2018) dataset (Xiao et al., 2018). The CAIL2018 dataset consists of a large of legal cases published by the Supreme People's Court of China and it has two sub-datasets, namely, CAIL-small and CAIL-big. Every case has a fact description and the judgment results given by human judges. The statistics of the dataset are shown in Table 2.

In addition, to be consistent with the baseline methods (Zhong et al., 2019; Xu et al., 2020; Yue et al., 2021), we first filter out the legal cases with multiple article/charge labels in the CAIL dataset, and then filter out the low-frequency law articles and charges which have less than 100 cases. Finally, we filter out the legal cases with missing or error labels (*e.g.* a small number of cases have no law article or charge labels, or the charge label is inconsistent with the law article label).

5.2 Baselines

In order to verify the effectiveness of our model, we select several representative legal judgment prediction models as the baselines.

(1) **FLA** (Luo et al., 2017) first considers the important role of law articles in the actual legal judgment process and uses the attention module to introduce the law article information.

(2) **Attribute-Att** (Hu et al., 2018) considers distinguishing the confusing charges is hard by introducing brief and concise law articles, and then designs ten common artificial attributes for charges.

(3) **TOPJUDGE** (Zhong et al., 2018) first takes into account the sequence dependency of the three subtasks of legal judgment prediction in the actual scenario. This model designs a topological multi-task learning framework to use the dependency information.

(4) **MPBFN-WCA** (Yang et al., 2019) takes into account that the judge needs to check again whether the relevant law articles, charges and term of penalty are suitable.

(5) **LADAN** (Xu et al., 2020) considers distinguishing the confusing law articles and design a graph distillation operator to learn the differences among law articles.

(6) **Neurjudge** (Yue et al., 2021) takes into account the circumstances in the actual scenario and use the intermediate results to separate the fact description vector representation. It is one of the state-of-the-art models.

(7) **CR4LJP** is our method.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.

Method	Law Articles				Charges				Term of Penalty			
	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
FLA	0.8853	0.8463	0.8067	0.8188	0.8732	0.8414	0.8134	0.8119	0.3566	0.3279	0.3176	0.3104
Attribute-Att	0.8910	0.8490	0.8357	0.8396	0.8896	0.8587	0.8343	0.8450	0.3686	0.3355	0.3288	0.3246
TOPJUDGE	0.8940	0.8578	0.8348	0.8430	0.8819	0.8513	0.8331	0.8379	0.3668	0.3296	0.3494	0.3275
MPBFN-WCA	0.8944	0.8600	0.8434	0.8478	0.8820	0.8537	0.8393	0.8425	0.3677	0.3417	0.3346	0.3357
LADAN	0.9016	0.8711	0.8556	0.8604	0.8871	0.8588	0.8451	0.8464	0.3718	0.3496	0.3488	0.3383
Neurjudge	0.9112	0.8853	0.8661	0.8720	0.8913	0.8663	0.8486	0.8512	**0.4064**	**0.3780**	**0.3641**	**0.3656**
CR4LJP	**0.9137**	**0.8868**	**0.8785**	**0.8791**	**0.8932**	**0.8675**	**0.8570**	**0.8596**	0.3802	0.3714	0.3398	0.3431

Table 3: Results with GRU-based encoder on CAIL-small dataset. The best results are in bold.

Method	Law Articles				Charges				Term of Penalty			
	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
FLA	0.9436	0.8471	0.7870	0.8091	0.9383	0.8390	0.7765	0.7993	0.5338	0.4223	0.4033	0.4097
Attribute-Att	0.9512	0.8787	0.7849	0.8137	0.9469	0.8759	0.7821	0.8148	0.5503	0.4552	0.3941	0.4126
TOPJUDGE	0.9502	0.8648	0.8021	0.8246	0.9461	0.8643	0.7943	0.8201	0.5574	0.4583	0.4040	0.4206
MPBFN-WCA	0.9507	0.8733	0.8054	0.8291	0.9457	0.8656	0.7957	0.8189	0.5583	0.4429	0.4110	0.4221
LADAN	0.9530	0.8719	0.8141	0.8345	0.9427	0.8607	0.8070	0.8263	0.5799	0.4833	0.4334	0.4413
Neurjudge	0.9568	0.8841	0.8307	0.8497	0.9505	0.8707	0.8197	0.8356	**0.5805**	0.4851	**0.4611**	**0.4638**
CR4LJP	**0.9594**	**0.8850**	**0.8449**	**0.8576**	**0.9524**	**0.8800**	**0.8235**	**0.8436**	0.5801	**0.4864**	0.4537	0.4560

Table 4: Results with GRU-based encoder on CAIL-big dataset. The best results are in bold.

5.3 Experiment Setting

For the GRU-based encoder, we first use the tool THULAC (Sun et al., 2016) to do word segmentation for the fact description and pre-train the word embedding with the dimension of 200 using word2vec (Mikolov et al., 2013). The maximum text length of the fact description is set to 400 for all the models. The hidden size is set to 150 for all the models. The learning rate is set to 1e-3. For the Bert-based encoder, the learning rate is set to 1e-5. Our model is trained on one V100 GPU (2 V100 GPUs for Bert-based encoder) for 20 epochs and the batch size is 128. We set the hyperparameter α_i to 1 for three subtasks and we use the AdamW optimizer to train our model. For the case retrieval module, we select the same number of 20 cases for each charge under each law article and set the k of Top-k as 5. We use Accuracy (Acc.), Macro Precision (MP), Macro Recall (MR), and Macro F1 (F1) to measure all models following the previous works.

5.4 Overall Results

The experimental results of all the models on three legal judgment prediction subtasks are shown in Table 3 and Table 4. Compared with the best baseline model Neurjudge, our model CR4LJP increases F1 scores of the law article and charge prediction subtasks by 0.81% and 0.98% respectively on the CAIL-small dataset and increases F1 scores of these two subtasks by 0.93% and 0.96% respectively on CAIL-big dataset. The experimental results prove the effectiveness of the model. It should be noted that the results of the term of penalty prediction task of our model on the CAIL-small and CAIL-big datasets are still worse than those of Neurjudge. Neurjudge simulates the actual judicial process and makes fine-grained division of the case description which is based on human knowledge and proved very effective for the term of penalty prediction.

Compared with the results of other baseline models, we can draw the following conclusions:

(1) TOPJUDGE and MPBFN-WCA both make use of the dependencies among the three subtasks to improve the fact representation of a single case. The better results of our model show that retrieving

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.

Method	Law Articles				Charges				Term of Penalty			
	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
Bert	0.9238	0.8987	0.8822	0.8859	0.9139	0.8897	0.8759	0.8792	0.4083	0.3837	0.3486	0.3425
Bert-Crime	0.9235	0.8948	0.8875	0.8872	0.9145	0.8898	0.8844	0.8838	0.4100	0.4013	0.3409	0.3441
Neurjudge+Bert	0.9314	0.9112	0.9041	0.9064	0.9230	0.9065	0.8994	0.9010	**0.4126**	**0.3977**	**0.3594**	**0.3670**
CR4LJP+Bert	**0.9343**	**0.9140**	**0.9043**	**0.9070**	**0.9245**	**0.9067**	**0.9007**	**0.9022**	0.4072	0.3857	0.3447	0.3501

Table 5: Results with Bert-based encoder on CAIL-small dataset.

historical cases help the model get a better representation of the fact to improve the performance of judgment results.

(2) The performance of FLA is worse than those of other neural network models because it directly introduces the Top-k law articles, but the law articles are short and confusing, which may bring some noise. To solve the confusing law article problem, Attribute-Att designs ten artificial attributes and LADAN designs a graph distillation operator to improve the representation of introduced law articles. The better results of our model CR4LJP show that historical case information is more helpful to improve the performance of the legal judgment prediction task than law article information.

(3) For the results of all models on the relevant law article and charge prediction subtasks, the F1 scores on the CAIL-big dataset are worse than those on the CAIL-small dataset, and the accuracy is the opposite. The main reason is that the categories of law articles or charges in the CAIL-big dataset are more unbalanced than those in the CAIL-small dataset.

5.5 Results with Bert Based Encoder

The pre-trained language models achieve the best results on many NLP tasks, such as Bert(Devlin et al., 2019). These models can be used as encoders in our framework, which usually leads to better results. In order to show the flexibility of our model, we compare our model based on Bert encoder with other methods.

• **Bert** uses the fact description as the input and the output of "[CLS]" token as the representation of the fact description. We fine-tune it on the CAIL dataset for the legal judgment prediction subtasks.

• **Bert-Crime** (Zhong et al., 2019) pre-trains Bert on a larger legal dataset. The process of fine-tuning is the same as **Bert**.

• **Neurjudge+Bert** is the Bert-based Neurjudge model, which replaces the GRU encoder with Bert.

• **CR4LJP+Bert** is our Bert-based model.

Due to the limitation of computing resources and the huge amount of parameters of Bert, we only conduct experiments on the CAIL-small dataset. Specifically, on the CAIL-small dataset, the time required for one epoch of training CR4LJP+Bert is about 36 times that of GRU based model (41400s vs 1140s). The experimental results are shown in Table 5. CR4LJP+Bert achieves better results than the GRU-based encoder. This shows the flexibility and effectiveness of our framework.

Compared with other baseline Bert models, the following observations can be observed:

(1) Our model (CR4LJP+Bert) is better than Bert and Bert-Crime indicating that additional case information can improve the performance of the legal judgment prediction task.

(2) Our model is superior to Neurjudge+Bert, which proves once again the effectiveness of our case retrieval framework.

5.6 Ablation Study

In order to verify the effectiveness of the case retrieval module for three subtasks, we perform an ablation study. Specifically, we remove the Top-k historical cases' mean vector from each subtask to study the impact of the case retrieval module. The corresponding models are expressed as **w/o article**, **w/o charge**, and **w/o term**.

The ablation study results of CAIL-small dataset are shown in Figure 3. We can see:

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.

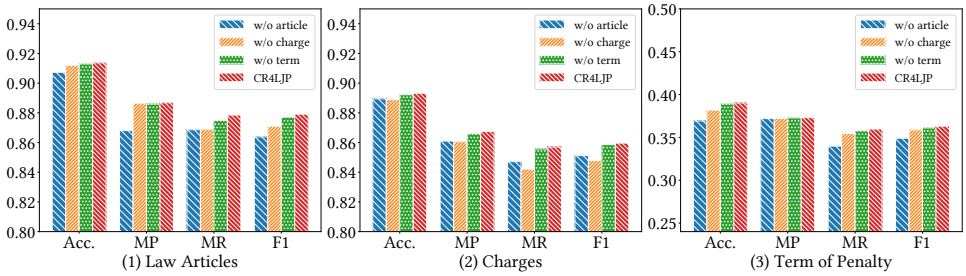

Figure 3: Ablation study on CAIL-small dataset. We remove the Top-k historical cases' mean vector from each subtask to study the impact of the case retrieval module.

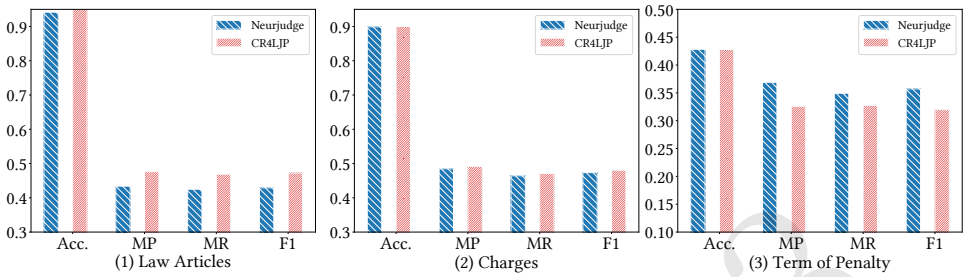

Figure 4: Case study on confusing charges.

(1) Removing the case retrieval module will degrade the performance of the three subtasks. This shows that the case retrieval module is effective.

(2) Removing the case retrieval module has the least impact on the term of penalty prediction subtask, which is in line with our expectations. In reality, the term of penalty prediction needs to be discussed and determined in more detail.

(3) In general, removing the case retrieval module from the law article prediction subtask has the most impact on the legal judgment prediction task. The underlying reason is that this subtask provides the basis for the other two tasks in reality, so it plays the most important role in the legal judgment prediction task.

5.7 Confusing Case study

As shown in Figure 1, Article #114 and Article #115 stipulate some similar charges. It is difficult to distinguish these cases with similar charge labels. In order to intuitively show the effect of models in distinguishing easily confusing cases, we select the cases related to Article #114 and Article #115 in the CAIL-small test set as a tiny dataset and test the baseline model Neurjudge and our model CR4LJP on the dataset.

From the experimental results in Figure 4, it can be seen that our model CR4LJP has better performance on the law article and charge prediction subtasks than Neurjudge, which shows that the case retrieval framework we proposed can effectively improve the ability to distinguish confusing cases.

6 Conclusion

In this paper, we first consider that judges usually look up some typical historical cases before making a judgment. We design a historical case retrieval model framework to simulate this scenario. For the current case, we retrieve the Top-k similar historical cases and get vector representation of these cases using the basic encoder, then we concatenate the mean vector of them to the fact description vector to predict the judgment results. Experimental results show that our method is effective.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.

Acknowledgements

We thank all the reviewers for their insightful comments. Zhicheng Dou is the corresponding author. This work was supported by National Key R&D Program of China No. 2022ZD0120103, National Natural Science Foundation of China No. 62272467, Beijing Outstanding Young Scientist Program No. BJJWZYJH012019100020098, Public Computing Cloud, Renmin University of China, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. The work was partially done at Beijing Key Laboratory of Big Data Management and Analysis Methods, and Key Laboratory of Data Engineering and Knowledge Engineering, MOE.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics.

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 *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 983–992. ACM.

Yi Feng, Chuanyi Li, and Vincent Ng. 2022. 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, May. Association for Computational Linguistics.

Anne von der Lieth Gardner. 1984. *An artificial intelligence approach to legal reasoning*. Ph.D. thesis, Stanford University.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search engine guided neural machine translation. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5133–5140. AAAI Press.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Few-shot charge prediction with discriminative legal attributes. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 487–498. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Fred Kort. 1957. Predicting supreme court decisions mathematically: A quantitative analysis of the "right to counsel" cases. *American Political Science Review*, 51(1):1–12.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1736–1745. Association for Computational Linguistics.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.

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.

Zhengyi Ma, Zhicheng Dou, Guanyue Bian, and Ji-Rong Wen. 2020. PSTIE: time information enhanced personalized search. In Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux, editors, *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 1075–1084. ACM.

Luyao Ma, Yating Zhang, Tianyi Wang, Xiaozhong Liu, Wei Ye, Changlong Sun, and Shikun Zhang. 2021. Legal judgment prediction with multi-stage case representation learning in the real court setting. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 993–1002. ACM.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Stuart S Nagel. 1963. Applying correlation analysis to case prediction. *Tex. L. Rev.*, 42:1006.

Jeffrey A Segal. 1984. Predicting supreme court cases probabilistically: The search and seizure cases, 1962-1981. *American Political Science Review*, 78(4):891–900.

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

S Sidney Ulmer. 1963. Quantitative analysis of judicial processes: Some practical and theoretical applications. *Law and Contemporary Problems*, 28(1):164–184.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018. R^3: Reinforced ranker-reader for open-domain question answering. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5981–5988. AAAI Press.

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.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2018. CAIL2018: A large-scale legal dataset for judgment prediction. *CoRR*, abs/1807.02478.

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.

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.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. Kg-fid: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4961–4974. Association for Computational Linguistics.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021. Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In *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.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.

Han Zhang, Zhicheng Dou, Yutao Zhu, and Jirong Wen. 2021. Few-shot charge prediction with multi-grained features and mutual information. In Sheng Li, Maosong Sun, Yang Liu, Hua Wu, Kang Liu, Wanxiang Che, Shizhu He, and Gaoqi Rao, editors, *Chinese Computational Linguistics - 20th China National Conference, CCL 2021, Hohhot, China, August 13-15, 2021, Proceedings*, volume 12869 of *Lecture Notes in Computer Science*, pages 387–403. Springer.

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.

Haoxi Zhong, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2019. Open chinese language pre-trained model zoo. Technical report.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does NLP benefit legal system: A summary of legal artificial intelligence. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5218–5230. Association for Computational Linguistics.

Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Encoding history with context-aware representation learning for personalized search. In Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu, editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1111–1120. ACM.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 801-812, Harbin, China, August 3 - 5, 2023.