Abstract

Given the fact description text of a legal case, legal judgment prediction (LJP) aims to predict the case’s charge, law article and penalty term. A core problem of LJP is how to distinguish confusing legal cases, where only subtle text differences exist. Previous studies fail to distinguish different classification errors with a standard cross-entropy classification loss, and ignore the numbers in the fact description for predicting the term of penalty. To tackle these issues, in this work, first, we propose a moco-based supervised contrastive learning to learn distinguishable representations, and explore the best strategy to construct positive example pairs to benefit all three subtasks of LJP simultaneously. Second, in order to exploit the numbers in legal cases for predicting the penalty terms of certain cases, we further enhance the representation of the fact description with extracted crime amounts which are encoded by a pre-trained numeracy model. Extensive experiments on public benchmarks show that the proposed method achieves new state-of-the-art results, especially on confusing legal cases. Ablation studies also demonstrate the effectiveness of each component.

1 Introduction

Artificial intelligence (AI) technologies have recently been applied to the legal domain to help legal practitioners reduce heavy and repeat work in many tasks, such as legal judgment prediction (Cui et al., 2022; Feng et al., 2022a; Zhong et al., 2020a), legal information extraction (Cardellino et al., 2017; Yao et al., 2022), legal summarization (Bhattacharya et al., 2019), and legal question answering (Zhong et al., 2020b; Martinez-Gil, 2021). Legal judgment prediction (LJP) is one of the most attractive research topics among these tasks. Given the fact description text of a legal case, legal judgment prediction (LJP) is aimed at predicting its charge, law article, and penalty term (Aletras et al., 2016; Zhong et al., 2018; Yang et al., 2019; Chalkidis et al., 2019).

One core problem hindering the performance of LJP from being satisfying is confusing legal cases, which have subtle text differences, but with totally different charges, law articles and penalty terms. A strand of studies on this problem has been conducted. Hu et al. (2018) manually annotate discriminative legal attributes for confusing charges and incorporate this explicit knowledge into an attention-based multi-task learning judgment prediction framework. Xu et al. (2020) explore to extract distinguish knowledge from similar law articles with a graph-based method. NeurJudge (Yue et al., 2021) utilizes the results of intermediate subtasks to separate the fact description into different circumstances and exploits them to make the predictions of other subtasks.

However, we argue that there exist two drawbacks to these studies. First, they all use a standard cross-entropy classification loss, which cannot distinguish different mistaken classification errors. For example, the error of classifying the charge Theft into Theft’s confusing charge Robbery is the same as classifying Theft into a not confusing charge Rape. The model needs to be punished more if it classifies a charge into its corresponding confusing charges. Second, the numbers in fact descriptions are crucial evidence for predicting the penalty terms of certain cases, such as financial cases. However, these numbers are processed equally with other words of the fact descriptions. It is hard for the model to deduce the total crime amount, and make correct penalty term predictions based on the scattered numbers.

To tackle these issues, in this work, we improve confusing legal judgment prediction via contrastive learning and numeracy evidence. In order to deal with the drawback of the cross-entropy classification loss, we propose to use supervised contrastive learning (Khosla et al., 2020; Gunel et al., 2020).
to pull fact representations from the same class closer and push apart fact representations from confusing charges. The use of supervised contrastive learning has been shown effective in many domains (Suresh and Ong, 2021; Zhang et al., 2022; Li et al., 2022). However, it is non-trivial to apply the original supervised contrastive learning to LJP. The reasons are two-fold. First, the number of charge classes is much larger than the number studied in (Gunel et al., 2020), which increases the difficulty to find sufficient negative examples in the mini-batches. Second, how to apply the original single-task supervised contrastive learning to the three subtasks of LJP remains explorations, especially when the subtasks labels are contradictory. The word “contradictory” here means that instances with the same charge labels may have different law labels or penalty term labels. If we pair the training instances with the same charge labels as positive examples, the resulting learned shared features will benefit the charge prediction task but will degrade the performance of the other two tasks. To tackle these challenges, we modify the momentum contrast (MoCo) unsupervised representation learning (He et al., 2020) into the supervised learning setting, and explore the best way to construct positive examples, which can benefit all the three subtasks of LJP simultaneously.

Second, in order to provide direct numerical evidence for the penalty term prediction, we need to extract the total crime amount from the fact descriptions and fuse the crime amount number into the penalty prediction model. The first challenge here is computing the total crime amount. Because the numbers scatter in the fact descriptions, only some numbers are part of the crime amount while the rest are not. To deduce the total crime amount of the legal case, we formalize the number extraction as a named entity recognition task, where the recognized numbers make up the final crime amount. Then, we consider fusing the crime amount number into the penalty prediction model while preserving the numeracy of the crime amount, which is achieved by a pre-trained number encoder model DICE (Sundararaman et al., 2020).

We conduct extensive experiments on public benchmarks (i.e., CAIL-small and CAIL-big). The results show that the proposed method achieves new state-of-the-art results, giving up to a 1.6 F1 score improvement for confusing legal cases and a 3.73 F1 score improvement for numerically sensitive legal cases. Ablation studies also demonstrate the effectiveness of each component.

2 Related Work

Legal Judgment Prediction. In recent years, with the increasing availability of public benchmark datasets (Xiao et al., 2018a; Ma et al., 2021; Feng et al., 2022b) and the development of deep learning, LJP has become one of the hottest topics in legal artificial intelligence (Yang et al., 2019; Zhong et al., 2020a; Cui et al., 2022; Feng et al., 2022a). Various deep learning techniques have been exploited for LJP, such as attention (Luo et al., 2017; Ma et al., 2021), graph neural network (Xu et al., 2020), multi-task learning (Zhong et al., 2018; Hu et al., 2018; Yue et al., 2021) and knowledge injection (Luo et al., 2017; Hu et al., 2018; Xu et al., 2020; Gan et al., 2021). Our work focuses on confusing legal judgment prediction, which is a difficulty in LJP. To solve this challenge, Hu et al. (2018) manually annotate discriminative legal attributes for confusing charges. Xu et al. (2020) explore to extract distinguish knowledge from similar law articles using a graph-based method. Neur-Judge (Yue et al., 2021) utilizes the results of intermediate subtasks to separate the fact description into different circumstances and exploits them to make the predictions of other subtasks.

Different from previous studies, our work propose to use supervised contrastive learning to better exploit the label information and addition extracted numerical evidence for penalty term prediction.

Contrastive Learning. The goal of contrastive learning (Chopra et al., 2005) is to learn distinguishing representations so that similar examples are closer in the feature space while dissimilar examples are far away in the feature space. Contrastive learning has been widely discussed in computer vision, especially in self-supervised settings (Wu et al., 2018; Chen et al., 2020; He et al., 2020). Recently, several studies have extended contrastive learning to supervised settings (Gunel et al., 2020; Khosla et al., 2020; Suresh and Ong, 2021; Zhang et al., 2022), where examples belonging to the same label in the mini-batch are regarded as positive examples to compute additional contrastive losses.

In this work, we explore how to apply this supervised contrastive learning, which is originally designed for single tasks, for the three subtasks of legal judgment prediction.
3 Method

3.1 A Multi-Task Learning Framework for Legal Judgment Prediction

Let \( f = \{ s_1, s_2, ..., s_N \} \) denotes the fact description of a legal case, where sentence \( s_i = \{ w_1, w_2, ..., w_M \} \) contains \( M \) words, and \( N \) is the number of sentence. Given a fact description \( f \), the LJP task aims at predicting its charge \( y_c \in \mathbb{C} \), law article \( y_l \in \mathbb{L} \) and term of penalty \( y_t \in \mathbb{T} \).

While previous studies have designed various neural architectures for LJP, these models can be boiled down to the following multi-task learning framework. The overall multi-task learning framework of LJP is shown in Figure 1. Specifically, first, a shared fact encoder is used to encode \( f \) into basic legal document representations.

\[
H_f = \text{SharedFactEncoder}(F)
\]

The shared fact encoder \( \text{SharedFactEncoder} \) could be Long-Short Term Memory Network (LSTM; (Hochreiter and Schmidhuber, 1997)), Gated Recurrent Unit (GRU; (Cho et al., 2014)) or pre-trained language models, e.g., BERT (Devlin et al., 2019).

Second, in order to learn task-specific representations for each subtask, three private encoders are designed to transform the shared representation \( H_f \) into specific task representations. For example, NeurJudge (Yue et al., 2021) utilizes the results of intermediate subtasks to separate the shared fact representations into different circumstances to improve the predictions of other subtasks. This process is formalized as follows.

\[
H_c = \text{ChargeEncoder}(H_f)
\]
\[
H_l = \text{LawEncoder}(H_f)
\]
\[
H_t = \text{TermEncoder}(H_f)
\]

It is worth noting that the subtasks do not share the parameters of these private encoders.

Third, based on these task-specific representations, different classification heads are used to compute the losses for the three tasks.

\[
L_c = \text{CrossEntropy}(\text{MLP}(H_c), y_c)
\]
\[
L_l = \text{CrossEntropy}(\text{MLP}(H_l), y_l)
\]
\[
L_t = \text{CrossEntropy}(\text{MLP}(H_t), y_t)
\]

where MLP is multi-layer perceptron and CrossEntropy is the cross-entropy classification loss, respectively.

The training objective is the sum of each task’s loss as follows:

\[
L_{ce} = L_c + L_l + L_t
\]
First, we propose MoCo-based supervised contrastive learning to overcome the difficulty of finding sufficient positive examples in LJP, which use the momentum update queue (He et al., 2020) as the look-up dictionary. Specifically, we maintain one feature queue \( Q \) to store sample features of interest, and one label queue \( L \) to store sample labels. The queue size can be much larger than a typical mini-batch size, thus can provide sufficient samples for computing the contrastive loss. Given \( Q \) and \( L \), for each example \( i \) in the mini-batch \( I \), according to the labels in \( L \), we can select positive and negative samples from \( Q \) to compute the supervised contrastive loss as follows:

\[
\mathcal{L}_{\text{sup}} = \sum_{i \in I} \frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(e_i \cdot e_p / t)}{\sum_{a \in A(i)} \exp(e_i \cdot e_a / t)}
\]  

(10)

where \( P(i) = \{q | y_q = y_i, q \in L \} \) and \( A(i) = \{q | q \neq i, q \in L \} \). \( e_i \) is the query feature encoded by a query encoder whose parameters are denoted as \( \theta_q \). The features in \( Q \) are encoded by a momentum key encoder whose parameters \( \theta_k \) are momentum updated as follows:

\[
\theta_k \leftarrow m\theta_{k-1} + (1-m)\theta_q
\]

(11)

where \( m \in [0, 1) \) is a momentum coefficient. Note that \( \theta_k \) are fixed during the back-propagation.

Next, based on the above MoCo-based supervised contrastive learning, we explore two strategies to perform multi-task supervised contrastive learning for LJP.

**Strategy I**  A straightforward strategy is to compute a contrastive loss for each task, and then sum them into one loss. Formally, three feature queues, i.e., \( Q^c, Q^l \) and \( Q^t \), are used to store task-specific feature, i.e., \( H_c, H_l \) and \( H_t \). Three label queues, i.e., \( L^c, L^l \) and \( L^t \) are used to store task labels, i.e., \( y_c, y_l \) and \( y_t \). Based on these denoted queues, we use Eq. 10 compute contrastive loss for each task and the weighted sum of them is denoted as the final loss:

\[
\mathcal{L}_{cl} = \alpha \mathcal{L}^c_{\text{sup}} + \beta \mathcal{L}^l_{\text{sup}} + \theta \mathcal{L}^t_{\text{sup}}
\]  

(12)

The final training objective of Strategy I is defined by:

\[
\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{cl}
\]

(13)

**Strategy II**  When closely examining the above supervised contrastive loss, we can find a severe contradictory phenomenon: summing the three contrastive losses together into one loss may not improve the overall performance. The reason is that while \( \mathcal{L}^c_{\text{sup}} \) will improve the performance of task \( i \), it may have negative effect on the other two tasks. For example, \( \mathcal{L}^c_{\text{sup}} \) view instances with the same charge labels as positive examples, however these instances may have different law labels or penalty term labels. As a result, \( \mathcal{L}^c_{\text{sup}} \) will force the shared encoder \( \text{SharedFactEncoder} \) to learn features that benefit charge prediction, but these shared features will degrade the performance of the other two tasks.

To solve this problem, we propose to view the instances whose three subtask labels are all the same as positive examples, and impose MoCo-based contrastive loss on the shared features \( H_f \). Specifically, we use a feature queue \( Q^B \) to store the shared features \( H_f \), and three label queues \( L^c, L^l \) and \( L^t \) to store the task labels. Then the positive samples set for sample \( i \) is denoted as \( P(i) = \{q | L^c(q) = y^c_i, L^l(q) = y^l_i, L^t(q) = y^t_i, q \in Q^B \} \). Based on this strategy, we can use Eq. 10 to compute the contrastive loss, denote as \( \mathcal{L}^B_{\text{sup}} \). Compared with Strategy I, Strategy II is able to overcome the contradictory phenomenon and improve the performance of all three subtasks.

The final training objective of Strategy II is defined by:

\[
\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}^B_{\text{sup}}
\]

(14)

### 3.3 Numerical Evidence Extraction as Named Entity Recognition

The numbers in legal cases are crucial evidence for predicting the penalty terms of specific charges, such as financial cases. However, the numbers are distributed randomly throughout the fact description and are processed the same as other words in earlier studies. Thus it is difficult for the model to directly deduce the accurate total crime amount and predict correct penalty terms based on the scattered numbers. In this work, we propose to provide direct numerical evidence for the penalty term prediction.

Towards this goal, first, we need a model to annotate the total crime amount for each instance in the LJP dataset. We formalize this crime amount annotation as a named entity recognition (NER) task, and the sum of the recognized numbers is regarded as the final crime amount. The reason for this formalization is that recognizing which numbers are part of the crime amount is a much simpler
Fact Description

The court found that at xxx, the defendants A and B stole a black rectangular wallet and a white cell phone (Metu M4 brand, worth $900) from the victim xxx, while he was asleep. The wallet contained cash of $2,600. After being arrested, the two defendants returned the aforementioned cell phone and cash of $2,000 to the victim.

Crime Amount: $3,500

Table 1: Data statistics.

| Dataset     | CAIL-Small | CAIL-Big |
|-------------|------------|----------|
| #Training Set Cases | 101,619 | 1,588,381 |
| #Validation Set Cases | 13,769 | 13,769 |
| #Test Set Cases | 26,749 | 185,290 |
| #Charges | 119 | 134 |
| #Law Articles | 103 | 121 |
| #Term of Penalty | 11 | 11 |

Figure 2: Numerical evidence extraction as named entity recognition.

Task than directly computing the total crime amount from the fact description.

Specifically, we train the crime amount tagging model on the LAC dataset\(^1\). Given an instance \((f, m)\) in LAC, where \(f\) and \(m\) denote fact description and crime amount respectively, we first convert the instance into the BIO NER format. The concrete format conversion process is as follows: we use a 0-1 package algorithm to find a set of sentences in \(f\), where the sum of their numbers equals to the crime amount \(m\). Then the numbers in the selected sentences are labeled as named entities. The converted dataset is named LAC-NER, based on which we train the state-of-art BERT-CRF as the NER model. Now, each instance in the LJP dataset can obtain a crime amount label \(m\) annotated by the well-trained NER model, denoted as \((f, y_c, y_t, y_l, m)\). Figure 2 shows an example of extracting crime amount from the fact description.

Second, we consider fusing the crime amount number into the penalty prediction model while preserving the numeracy of the crime amount. To achieve this goal, we propose to pre-train a number encoder model using DICE (Sundararaman et al., 2020). Given a pair of numbers, the cosine similarity of their DICE embeddings has a linear relationship with respect to their actual distance. Specifically, given a training data \(D = \{(x_i, y_i)\}_i^N\), where \((x_i, y_i)\) represents a pair of numbers, we use the following training objective to optimize the DICE encoder:

\[
\mathcal{L}_{num} = \| \frac{2|x_i - y_i|}{|x_i| + |y_i|} - \cos(x_i, y_i) \| \tag{15}
\]

where \(x_i\) and \(y_i\) are contextual representations of the two numbers, which are obtained by a LSTM model.

Given a training instance \((f, y_c, y_t, y_l, m)\), its crime amount \(m\) is encoded by the pre-trained DICE and then is fused into the penalty prediction head. Specifically, we modify Eq. (7) into the following formulas:

\[
m = \text{DICE}(m) \tag{16}
\]

\[
\mathcal{L}_t = \text{CrossEntropy}(\text{MLP}([H_t; m]), y_t) \tag{17}
\]

where \([; ;]\) denotes a concatenation operation.

4 Experiments

4.1 Datasets

To evaluate the effect of our method, we conduct experiments on two public benchmark datasets (i.e., CAIL-small and CAIL-big), from the Chinese AI and Law challenge (CAIL2018) (Xiao et al., 2018b). Each instance in both datasets contains one fact description, one applicable law article, one charge, and one term of penalty. To give a fair comparison, we use the code released by (Xu et al., 2020) to process the data. All models are trained on the same dataset. Table 1 shows the detailed statistics of the datasets.

4.2 Implementation Details

Following (Xu et al., 2020), we use the THULAC\(^2\) tool to segment Chinese into words. The word embedding layer in the neural network is initialized by pre-train word embeddings provided by (Zhong et al., 2018). The maximum document length and the maximum sentence number is set to 512 words and 15, respectively.

For pre-training the number encoder model DICE, we synthesize a training dataset of size 128,000 where each pair of numbers \((x_i, y_i)\) are uniformly sampled from \([0, 30, 0000]\). A GRU model with a hidden size of 256 is used to model the number sequence. Adam is used to optimize

\(^1\)http://data.court.gov.cn/pages/laic2021.html

\(^2\)https://github.com/thunlp/THULAC
Table 2: Main results on CAIL-small.

| Tasks | Charges | Law Articles | Term of Penalty |
|-------|---------|--------------|-----------------|
|       | Metrics | Acc. | MP | MR | F1 | Acc. | MP | MR | F1 | Acc. | MP | MR | F1 |
| HARNN | 84.54   | 82.56 | 82.94 | 82.26 | 80.09 | 76.46 | 77.69 | 75.95 | 38.38 | 36.12 | 33.99 | 34.32 |
| LADAN | 84.95   | 82.55 | 83.26 | 82.42 | 80.38 | 75.84 | 77.84 | 75.67 | 38.21 | 35.95 | 34.01 | 34.28 |
| NeurJudge+ | 83.25 | 82.11 | 81.69 | 81.3 | 80.95 | 77.93 | 78.59 | 77 | 37.88 | 37.2 | 33.82 | 34.92 |
| HARNN+NSCL | 85.26 | 83.93 | 83.76 | 83.39 | 81.07 | 77.95 | 78.59 | 77 | 39.18 | 37.32 | 34.5 | 35.03 |
| LADAN+NSCL | 85.37 | 83.91 | 84.04 | 83.57 | 81.32 | 78.06 | 78.59 | 77.24 | 39.38 | 37.95 | 35.23 | 35.95 |
| NeurJudge+NSCL | 84.45 | 83.3 | 83.55 | 82.88 | 81.12 | 78.1 | 78.98 | 77.32 | 39.65 | 39.48 | 34.65 | 36.22 |

Table 3: Main results on CAIL-big.

| Tasks | Charges | Law Articles | Term of Penalty |
|-------|---------|--------------|-----------------|
|       | Metrics | Acc. | MP | MR | F1 | Acc. | MP | MR | F1 | Acc. | MP | MR | F1 |
| HARNN | 96.48   | 88.1 | 83.54 | 85.34 | 96.54 | 84.88 | 79.42 | 81.4 | 60.30 | 51.08 | 47.47 | 48.79 |
| LADAN | 96.56   | 88.58 | 84.17 | 85.95 | 96.61 | 86.4 | 80.46 | 82.52 | 60.44 | 51.56 | 48.67 | 49.78 |
| NeurJudge+ | 95.48 | 85.57 | 79.55 | 81.49 | 96.26 | 85.78 | 81.38 | 82.8 | 58.40 | 49.67 | 43.32 | 44.90 |
| HARNN+NSCL | 96.55 | 88.84 | 83.96 | 85.82 | 96.60 | 86.63 | 81.16 | 82.99 | 60.20 | 52.18 | 47.50 | 49.16 |
| LADAN+NSCL | 96.65 | 89.34 | 84.57 | 86.49 | 96.71 | 87.60 | 81.74 | 83.68 | 60.56 | 51.85 | 48.86 | 49.81 |
| NeurJudge+NSCL | 95.73 | 86.37 | 80.88 | 82.58 | 96.20 | 85.48 | 81.56 | 82.78 | 58.40 | 49.10 | 43.54 | 45.43 |

Table 4: Effects of different supervised contrastive learning strategies.

| Methods | Charges | Law Articles | Term of Penalty |
|---------|---------|--------------|-----------------|
|       | Metrics | Acc. | F1 | Acc. | F1 | Acc. | F1 |
| LADAN | 83.25 | 82.42 | 80.38 | 75.87 | 38.21 | 34.28 |
| LADAN+I | 84.96 | 84.77 | 80.79 | 76.40 | 37.64 | 34.43 |
| LADAN+II | 86.01 | 83.83 | 80.97 | 77.04 | 39.50 | 35.71 |
| LADAN+I+II | 85.21 | 83.47 | 81.27 | 77.38 | 39.70 | 35.77 |

distinguishable features from similar law articles using a graph-based method.

- **NeurJudge+** (Yue et al., 2021): NeurJudge+ utilizes the results of intermediate subtasks to separate the fact description into different representations and exploits them to make the predictions of other subtasks.

We also evaluate the effects of the proposed method on pre-trained language models.

- **BERT** (Devlin et al., 2019): BERT is a pre-trained language model, which has achieved results for numerous tasks. As the evaluated datasets are in Chinese, we use bert-base-chinese to conduct experiments.

- **CrimeBERT** (Zhong et al., 2019): CrimeBERT is initialized by BERT, then is further pre-trained on crime data.

4.4 Development Experiments

To empirically evaluate which strategies is better for performing multi-task supervised contrastive learning for LJP, we conduct development experiments on CAIL-small. The adopted backbone is LADAN. The results are listed in Table 4.

From Table 4, we have the following observations. First, both Strategy I and II can improve

---

3We also try other settings for pre-training DICE, but they did not give better results.

4https://github.com/google-research/bert
LADAN on the three tasks. However, the gains of Strategy I is much smaller than the gains of Strategy II, which verifies our hypothesis that directly imposing contrastive learning loss $L^i_{sup}$ for each task will improve the performance of task $i$, but it has negative effect on the other two tasks. Second, we also explore the effect of combining these two strategies, i.e., using $L_{cl} + L_B$ as the supervised contrastive loss. As seen, the improvement of this combination is not significant.

Thus, in the rest experiments, we take Strategy II as the supervised contrastive learning loss if not stated otherwise. The final method combined supervised contrastive learning and numerical evidence is denoted as NSCL.

### 4.5 Main Results

To evaluate the effectiveness of the proposed method, we argument each baseline with the proposed two methods (i.e., NSCL) and conduct experiments on CAIL-small and CAIL-big. The results are listed in Table 2 and Table 3, from which we have the following observations.

First, the proposed method can improve all the baselines and achieve new state-of-the-art results on the two datasets. Specifically, on CAIL-small, the absolute improvements can be up to a 1.15 F1 score on the charge prediction, a 1.57 F1 score on the law article prediction and a 1.67 F1 score on the term of penalty prediction, respectively. Second, on CAIL-big, the gains are relatively small, giving an absolute 0.54 F1 score improvement on the charge prediction, an absolute 1.16 F1 score improvement on the law article prediction, and an absolute 0.53 F1 score improvement on the term of penalty prediction, respectively. Third, we observe that on CAIL-big, NeurJudge+ gives worse performance than other baselines. We suppose the reason is that the complex neural networks of NeurJudge+ cause its overfitting on CAIL-big.

We also conduct experiments on BERT and its variant CrimeBERT. As shown in Table 5, we can observe that: (1) CrimeBERT and BERT give better results than non pre-trained models, which demonstrates the advantages of pre-training; (2) moco-based supervised contrastive and numerical evidence can still improve pre-trained models on the three tasks; (3) Numerical evidence is particularly vital for number-sensitive legal cases, giving larger F1 score improvements.

### 4.6 Analysis

#### Effect of Contrastive Learning for Confusing Charges

We conduct experiments to validate the effect of contrastive learning for predicting charges, especially confusing charges. The concrete confusing charges we defined are listed in the Appendix. As shown in Table 7, the absolute F1 value improvements of confusing charges are greater than that of the overall charges. In particular, NeurJudge+...
Fact Descriptions: At 8 am on June 25, 2009, the defendant A fought with the victim B due to trivial matters in a market in xxx Town, xxx City, and wanted to take revenge on the victim B. At 8 pm on June 27 of the same year, the defendant A gathered three people including C, D and E, and fled to the open-air parking lot next to a market. The defendant A slashed the victim B's right arm with a watermelon knife, then they fled the scene. According to the forensic identification, the victim B suffered minor injuries.

Golden Labels: Charge 111; Law Article 34; Term of Penalty: 6
LADANMTL: Charge 1; Law Article 34; Term of Penalty: 4
LADANMTL+NSCL: Charge 111; Law Article 34; Term of Penalty: 6

Fact Descriptions: From November 8, 2015 to January 29, 2016, the defendant A committed three thefts in xxx City, with a total value of 10,400 yuan. 1. At 8 am on November 8, 2015, the defendant A stole a "Dayang brand" electric tricycle worth 5,600 yuan from the victim B. 2. On November 28, 2015, the defendant A stole a "Dayang" electric tricycle worth 4,800 yuan from the victim Gao in xxx city. 3. At 10:00 p.m. on January 29, 2016, the defendant A stole a electric vehicle battery from the victim C, but was discovered and attempted.

Golden Labels: Charge 110; Law Article 32; Term of Penalty: 7
LADANMTL: Charge 110; Law Article 56; Term of Penalty: 10
LADANMTL+NSCL: Charge 110; Law Article 56; Term of Penalty: 7; Extracted Money: 20,800

Table 8: Case studies.

obtains an absolute 2.55 F1 score improvement on confusing charges, which shows that the moco-based supervised contrastive learning is effective for learning distinguishable representations for confusing charges.

Effect of Numerical Evidence for Number-Sensitive Legal Cases To investigate the effect of numerical evidence for predicting the term of penalty, we conduct ablative experiments. As shown in Table 6, the improvement of moco-based contrastive learning only is relatively tiny. For example, it only gives 0.11 and 0.05 F1 score improvements for HARNN and NeurJudge+ on number-sensitive examples, respectively. However, if further providing the model with the extracted numerical evidence, the F1 scores of all the baselines have a considerable boost. In particular, combined with numerical evidence, LADANMTL obtains a 2.02 F1 score improvement on number-sensitive examples. These results show that the extracted crime amount is more beneficial than supervised contrastive learning for number-sensitive legal cases.

Visualization We use t-SNE (Van der Maaten and Hinton, 2008) to visualize the features used for prediction charge Pick a Quarrel, Intentional Injury and Crowd Fight. These charges are confusing to each other. As shown in Figure 3, compared with LADANMTL, the learned features (e.g., the features of charge Pick a Quarrel) of Strategy II are more compact. However, the features of charge Intentional Injury with Strategy I are less compact than the corresponding learned features of Strategy II, which demonstrates the advantage of Strategy II.

Case Study Table 8 shows two cases to demonstrate the effect of the proposed method. As shown in the first case, LADANMTL wrongly predicts the case’s charge, which should be Intentional Injury, into its confusing charge Pick a Quarrel. However, with the proposed contrastive learning, this error is corrected. The second case demonstrates the effect of numerical evidence. With the extracted crime amount of $20,800, the model is able to correctly predict the term of penalty.
5 Conclusion

In this work, firstly, we explore to use the supervised contrastive learning to learn distinguishable representations for confusing legal judgment predictions. Secondly, we provide direct numerical evidence for predicting the term of penalty. The extraction of the crime amounts is formalized as a named entity recognition task, and the extracted number evidence is fused into the representation of the fact description with a pre-trained numeracy model. Extensive experiments on public benchmark datasets show that the proposed method can improve existing LJP models, giving new state-of-the-art results. Ablation studies further demonstrate the effectiveness of the proposed method on confusing and number-sensitive legal cases.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoteiciu-Pietro, and Vasilieos Lampos. 2016. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. PeerJ Computer Science, 2:e93.

Paheli Bhattacharya, Kaustubh Hiware, Subham Rajgaria, Nilay Pochhi, Kripabandhu Ghosh, and Saptarshi Ghosh. 2019. A comparative study of summarization algorithms applied to legal case judgments. In European Conference on Information Retrieval, pages 413–428. Springer.

Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. 2017. Legal nerc with ontologies, wikipedia and curriculum learning. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 254–259.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in english. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4317–4323.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 103–111.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA, pages 539–546. IEEE Computer Society.

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. arXiv preprint arXiv: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, Volume 1 (Long and Short Papers), pages 4171–4186.

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. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

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.

Leilei Gan, Kun Kuang, Yi Yang, and Fei Wu. 2021. Judgment prediction via injecting legal knowledge into neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 12866–12874.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. In International Conference on Learning Representations.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9729–9738.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation, 9(8):1735–1780.

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, pages 487–498.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinetz, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. Advances in Neural Information Processing Systems, 33:18661–18673.
Mengze Li, Tianbao Wang, Haoyu Zhang, Shengyu Zhang, Zhou Zhao, Wenqiao Zhang, Jiaxu Miao, Shiliang Pu, and Fei Wu. 2022. Hero: Hierarchical spatio-temporal reasoning with contrastive action correspondence for end-to-end video object grounding. In Proceedings of the 30th ACM International Conference on Multimedia, pages 3801–3810.

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.

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 Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 993–1002.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. Journal of machine learning research, 9(11).

Jorge Martinez-Gil. 2021. A survey on legal question answering systems. arXiv preprint arXiv:2110.07333.

Dhanasekar Sundararaman, Shijing Si, Vivek Subramanian, Guoyin Wang, Devamanyu Hazarika, and Lawrence Carin. 2020. Methods for numeracy-preserving word embeddings. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 4742–4753. Association for Computational Linguistics.

Varsha Suresh and Desmond Ong. 2021. Not all negatives are equal: Label-aware contrastive loss for fine-grained text classification. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4381–4394.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3733–3742.

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

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018b. Cail2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv: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, pages 3086–3095.

Wenmian Yang, Weijia Jia, Xiaojie Zhou, and Yutao Luo. 2019. Legal judgment prediction via multi-perspective bi-feedback network. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pages 4085–4091.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Feng Yao, Chaojun Xiao, Xiaozhi Wang, Zhiyuan Liu, Lei Hou, Cunchao Tu, Juanzhi 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, pages 183–201.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021. Neurjudge: a circumstance-aware neural framework for legal judgment prediction. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 973–982.

Shu Zhang, Ran Xu, Caiming Xiong, and Chetan Ramaiah. 2022. Use all the labels: A hierarchical multi-label contrastive learning framework. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16660–16669.

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, pages 3540–3549.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020a. How does nlp benefit legal system: A summary of legal artificial intelligence. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5218–5230.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020b. Jecqa: a legal-domain question answering dataset. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 9701–9708.
Haoxi Zhong, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2019. Open chinese language pre-trained model zoo. Technical report.