Augmenting Legal Judgment Prediction with Contrastive Case Relations

Dugang Liu[1,2], **Weihao Du**[1,2], **Lei Li**[3], **Weike Pan**[1,2,*], **Zhong Ming**[1,2,*]
[1]Shenzhen University, Shenzhen, China
[2]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China
[3]Hong Kong Baptist University, Hong Kong, China
dugang.ldg@gmail.com, {panweike,mingz}@szu.edu.cn

Abstract

Existing legal judgment prediction methods usually only consider one single case fact description as input, which may not fully utilize the information in the data such as case relations and frequency. In this paper, we propose a new perspective that introduces some contrastive case relations to construct case triples as input, and a corresponding judgment prediction framework with case triples modeling (CTM). Our CTM can more effectively utilize beneficial information to refine the encoding and decoding processes through three customized modules, including the case triple module, the relational attention module, and the category decoder module. Finally, we conduct extensive experiments on two public datasets to verify the effectiveness of our CTM, including overall evaluation, compatibility analysis, ablation studies, analysis of gain source and visualization of case representations.

1 Introduction

As an important component of legal intelligence in civil law systems, legal judgment prediction (LJP) has received a lot of attention and research in recent years (Chalkidis et al., 2019; Zhong et al., 2020). Given a case fact description, LJP usually includes three sub-tasks, i.e., law article prediction, charge prediction and terms of penalty prediction for this case (Xiao et al., 2018), and an example of LJP is shown on the left side of Figure 1. As an auxiliary tool to serve legal practitioners and people without professional knowledge in law, a more accurate method for LJP is necessary.

The existing legal judgment prediction methods mainly include two lines of single-task modeling and multi-task modeling. The former usually focuses on targeted modeling of a certain sub-task, such as introducing some more advanced network architectures (Chen et al., 2019a; Le et al., 2020) or more sources of information (Luo et al., 2017; Hu et al., 2018; Chen et al., 2019b). The latter takes multiple sub-tasks as a whole and uses a multi-task learning (MTL) framework for unified modeling. The most representative methods in this line aim to design different decoding structures, including MTL (Zhong et al., 2018) that ignores the inter-task dependency, TopJudge (Zhong et al., 2018) that considers unidirectional topological dependency among sub-tasks, and MPBFN (Yang et al., 2019) that considers bidirectional topological dependency. In this paper, we focus on the line of multi-task learning because it is more aligned with practical applications.

Although the existing methods have shown promising results, as shown on the left side of Figure 1, most of them only consider the fact description of one single case as input when modeling. This form of modeling ignores the full utilization of the beneficial information contained in the data, such as the case relation and frequency information that might provide constraints for modeling. We believe this may have an adverse effect on the model and cause a performance bottleneck, such as cases with low-frequency law articles or charges suffer from insufficient training. As an example, we show in Figure 2 the accuracy of MPBFN on CAIL-small (Xiao et al., 2018) for law articles and charges of different frequencies. We can find that the accuracy drops significantly with decreasing frequency.

To more effectively utilize the beneficial information contained in the data, in this paper, we propose a new perspective that introduces some contrastive case relations to construct case triples as the input of the model. Specifically, we sample some similar and dissimilar cases for a current case through some carefully designed contrasting case relations, where these auxiliary cases will be beneficial to improve the performance of the model. An example of this new form of modeling is shown on

[*] Co-corresponding authors

Proceedings of the 29th International Conference on Computational Linguistics, pages 2658–2667
October 12–17, 2022.

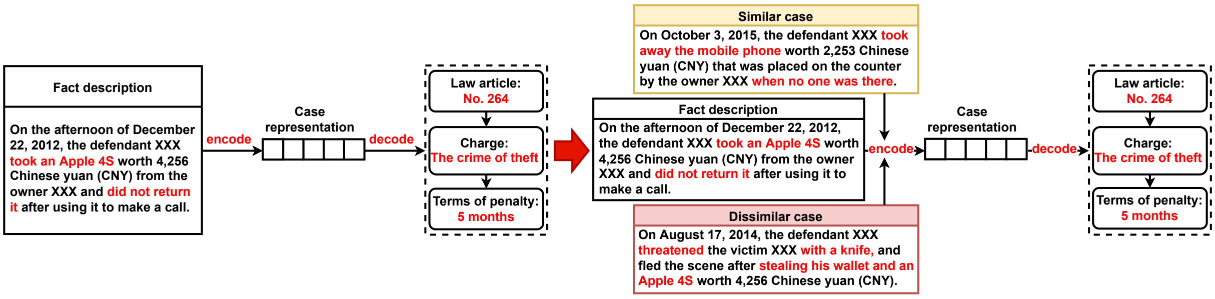

Fact description
On the afternoon of December 22, 2012, the defendant XXX took an Apple 4S worth 4,256 Chinese yuan (CNY) from the owner XXX and did not return it after using it to make a call.
encode
Case representation
decode
Law article: No. 264
Charge: The crime of theft
Terms of penalty: 5 months
Similar case
On October 3, 2015, the defendant XXX took away the mobile phone worth 2,253 Chinese yuan (CNY) that was placed on the counter by the owner XXX when no one was there.
Fact description
On the afternoon of December 22, 2012, the defendant XXX took an Apple 4S worth 4,256 Chinese yuan (CNY) from the owner XXX and did not return it after using it to make a call.
Dissimilar case
On August 17, 2014, the defendant XXX threatened the victim XXX with a knife, and fled the scene after stealing his wallet and an Apple 4S worth 4,256 Chinese yuan (CNY).
encode
Case representation
decode
Law article: No. 264
Charge: The crime of theft
Terms of penalty: 5 months

Figure 1: On the left is an illustrative example of legal judgment prediction (LJP), including a case fact description and the corresponding three prediction sub-tasks. On the right is an example of the proposed new form of modeling, where some contrastive case relations are introduced to construct case triples as input. Note that in this paper we focus on prediction of law articles and charges, since that of terms of penalty is known of high difficulty and variance.

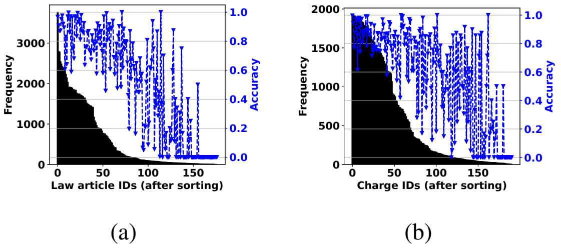


Figure 2: The prediction accuracy of MPBFN on CAIL-small for each law article and charge. Note that the IDs on the horizontal axis have been sorted in a descending order of frequency.

the right side of Figure 1, and the importance of different phrases may be more accurately identified through this case triple.

We then propose a corresponding judgment prediction framework with case triple modeling (CTM) to mine information from the constructed case triples for improving the model. Specifically, our CTM adds three customized modules to the traditional encoder and decoder: 1) a case triple module samples two similar cases and one dissimilar case for an input case based on case labels and frequency information to form two case triples; 2) a relational attention module imposes a relational constraint on the obtained case triples to refine the encoding process; and 3) a category decoder module acts as a switch to select a corresponding decoding branch for a high-frequency or low-frequency case to further refine the decoding process.

It is intuitive that our CTM does not depend on a specific encoder or decoder. This means that our CTM can be easily integrated with some existing legal judgment prediction methods, and we will demonstrate its good compatibility in the experiments by combining our CTM with different encoder and decoder structures. In addition to this, we conduct other empirical studies on two public datasets to verify the effectiveness of our CTM, including overall performance evaluation, ablation studies, fine-grained performance evaluation and case representation analysis.

2 Related Work

In this section, we briefly review some related works on two research topics, including legal judgment prediction and case relations modeling.

Legal Judgment Prediction. Legal judgment prediction can be mainly summarized into two research lines. The first line focuses on the targeted modeling of a specific sub-task from the perspective of network architectures (Chen et al., 2019a; He et al., 2019; Le et al., 2020), available information sources (Luo et al., 2017; Hu et al., 2018; Chen et al., 2019b), and interpretability of the models (Jiang et al., 2018). The second line considers multiple sub-tasks as a whole and uses a multi-task learning framework for case modeling. The most representative methods are MTL (Zhong et al., 2018), TopJudge (Zhong et al., 2018) and MPBFN (Yang et al., 2019), in which three different decoding structures are considered respectively. Some recent works have designed some more sophisticated architectures based on them, especially in combination with some graph learning techniques and large-scale pre-trained models (Xu et al., 2020; Chen et al., 2020; Dong and Niu, 2021; Yue et al., 2021). Note that since the terms of penalty prediction is usually of higher difficulty and variance than the other two sub-tasks, we focus on law article prediction and charge prediction similar to (Bao et al., 2019; Chen et al., 2021).

Case Relations Modeling. The idea of case rela-

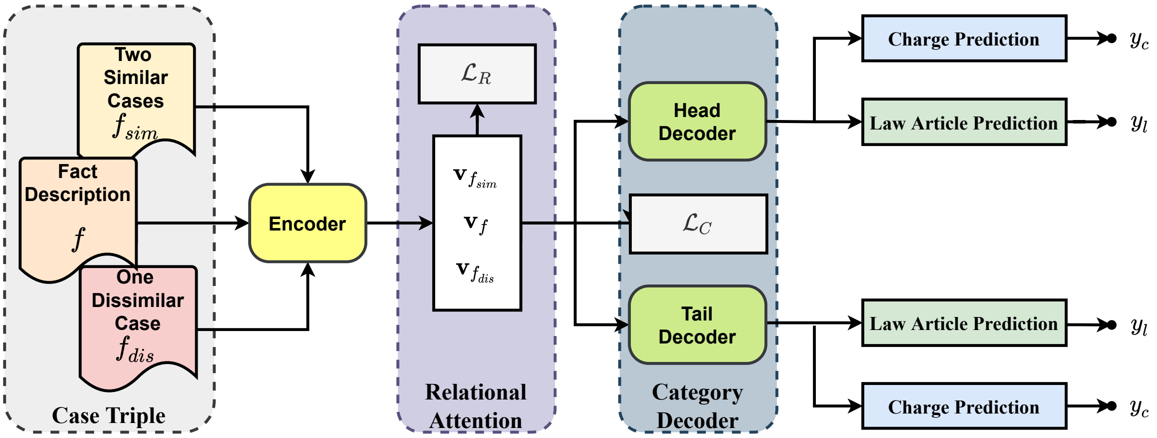


Figure 3: The architecture of a judgment prediction framework with our case triple modeling (CTM).

tions modeling is mainly applied to similar case matching (SCM) tasks in some recent studies on legal intelligence (Xiao et al., 2019; Peng et al., 2020; Hong et al., 2020). Unlike legal judgment prediction, this task is given a set of manually labeled case triples as training samples, where each triple contains two similar cases and one dissimilar case, and the goal is to learn a model that can identify those two similar ones. This task can be further relaxed to find some similar cases for a current case (Tang and Clematide, 2021; Ostendorff et al., 2021), which is important in the common law system. To the best of our knowledge, our work is the first to introduce a case triple structure to legal judgment prediction based on some case relations.

3 The Proposed Framework

3.1 Architecture

The judgment prediction framework with case triple modeling, or CTM for short, is shown in Figure 3. Note that similar to most works, we consider each case with only one law article label and one charge label for simplicity. Given a current case $f = [s; y_l, y_c, y_a]$, where $s = \{s_1, s_2, \ldots, s_n\}$ represents the fact description composed of sentences, y_l is the law article label, y_c is the charge label, and $y_a \in \{0, 1\}$ is a category label indicating whether the case is a high-frequency case or not. Note that a more specific description of high-frequency cases can be found in the case triple module in Sec 3.2. The case triple module samples two similar cases and one dissimilar case to construct the case triple (f, f_{sim}, f_{dis}) based on some contrastive case relations. Then, a constraint is imposed on the encoded representations corresponding to the case triple (i.e., $\mathbf{v}_f$, $\mathbf{v}_{f_{sim}}$ and $\mathbf{v}_{f_{dis}}$) in the relational attention module to refine the encoding process.

In the category decoder module, we first impose a classification constraint on the category label y_a to inform the model to which category the current encoded representation belongs, and then switch the decoder of the corresponding category branch to refine the decoding process. Finally, the model obtains the predicted label of each sub-task and compares it with the respective true label. The final optimization objective function of our CTM can be expressed as follows,

$$\min_{\theta} \mathcal{L}_{CTM} = \mathcal{L}_M + \mathcal{L}_R + \mathcal{L}_C + \lambda\|\theta\|, \quad (1)$$

where $\mathcal{L}_M$, $\mathcal{L}_R$, and $\mathcal{L}_C$ denote the prediction loss for multi-task learning, the constraint loss for the relational attention module and the loss for the category decoder module, respectively, and λ and $\|\theta\|$ are the tradeoff parameter and the regularization terms.

3.2 Training

In this section, we describe each module in detail based on the training process.

The Case Triple Module. We propose a concept called contrastive case relation that considers both labels and frequency information for constructing some case triples. Specifically, we use a threshold ϕ to pre-divide the labels of the law articles (and charges) into two sets of low-frequency $\mathcal{A}_l$ (or $\mathcal{A}_c$) and high-frequency $\mathcal{B}_l$ (or $\mathcal{B}_c$), where $\mathcal{A}_l$ (or $\mathcal{A}_c$) contains the labels with the lowest ϕ frequency and $\mathcal{B}_l$ (or $\mathcal{B}_c$) contains the remaining labels. For a case f, a similar case f^l_{sim} (or f^c_{sim}) on the law articles (or charges) is sampled from the candidate cases with the same law article (or charges) label. Then, a dissimilar case f_{dis} on the charges is sampled from the candidate cases with different charge labels and the corresponding labels do not belong to $\mathcal{A}_c$. The additional constraint that the labels do not belong to $\mathcal{A}_c$ help cases with low-frequency charge labels to be more fully trained based on a large number of opposite references. Since the law articles can be regarded as the leaf nodes of charges in civil law systems, i.e., different charge labels must have different law article labels, we regard this dissimilar case on the charges as a shared dissimilar case, i.e., it is also regarded as a dissimilar case on the law article. This can reduce the number of cases that need to be encoded in a subsequent fact description encoder module to reduce the size of the model. Finally, we can obtain two types of case triples (f, f^l_{sim}, f_{dis}) and (f, f^c_{sim}, f_{dis}) for f.

Considering that when f is a high-frequency case, i.e., $y_l \in \mathcal{B}_l$ or $y_c \in \mathcal{B}_c$, the above two case triples can enhance the distinction between the high-frequency cases. When f is a low-frequency case, i.e., $y_l \in \mathcal{A}_l$ or $y_c \in \mathcal{A}_c$, these triples can improve its insufficient training and enhance the distinction between it and the high-frequency cases by introducing a large number of high-frequency cases as opposite references. For ease of understanding, we give an example of the sampling process in Figure 4.

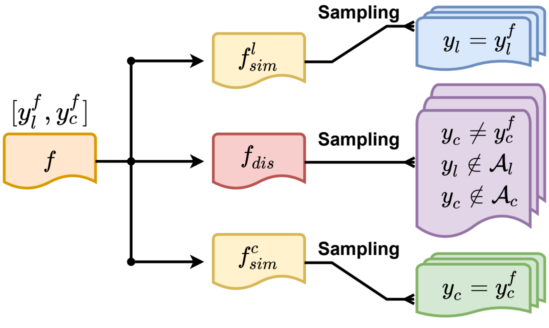


Figure 4: The schematic diagram of a sampling process, where the law article and charge labels of a case f are assumed to be y_l^f and y_c^f, respectively.

The Fact Description Encoder Module. After constructing the case triples, we need to encode the fact description of each case. Next, we use hierarchical Bi-GRU (Yang et al., 2016) as an example encoder[1], which has also been adopted in some recent works (Long et al., 2019; Xu et al., 2020; Ma et al., 2021). Specifically, let each sentence in the fact description be represented as $s_i = [w_{i,1}, w_{i,2}, \ldots, w_{i,m}]$, where $w_{i,j}$ represents the j-th word of sentence s_i, and m denotes the number of words, a word-level Bi-GRU will act on each sentence and output a corresponding representation (Yang et al., 2016),

$$\mathbf{h}_{i,j} = [\overrightarrow{\mathrm{GRU}}(\mathbf{w}_{i,j}), \overleftarrow{\mathrm{GRU}}(\mathbf{w}_{i,j})] \in \mathbb{R}^{d_w},$$

$$\alpha_{i,j} = \frac{\exp(\tanh(\mathbf{W}_w \mathbf{h}_{i,j} + \mathbf{b}_w)^\mathsf{T} \mathbf{u}_w)}{\sum_j \exp(\tanh(\mathbf{W}_w \mathbf{h}_{i,j} + \mathbf{b}_w)^\mathsf{T} \mathbf{u}_w)},$$

$$\mathbf{v}_{s_i} = \sum_{j=1}^{m} \alpha_{i,j} \mathbf{h}_{i,j},$$

where $\mathbf{w}_{i,j}$ represents an embedding vector of word $w_{i,j}$, $\mathbf{W}_w \in \mathbb{R}^{d_w \times d_w}$ is a weight matrix, $\mathbf{b}_w \in \mathbb{R}^{d_w}$ is a bias vector and $\mathbf{u}_w \in \mathbb{R}^{d_w}$ is a trainable context vector. Then, a sentence-level Bi-GRU will act on the representation sequence of the sentences, i.e., $[\mathbf{v}_{s_1}, \mathbf{v}_{s_2}, \ldots, \mathbf{v}_{s_n}]$, to obtain the encoded representation of case f (Yang et al., 2016),

$$\mathbf{h}_i = [\overrightarrow{\mathrm{GRU}}(\mathbf{v}_{s_i}), \overleftarrow{\mathrm{GRU}}(\mathbf{v}_{s_i})] \in \mathbb{R}^{d_s},$$

$$\alpha_i = \frac{\exp(\tanh(\mathbf{W}_s \mathbf{h}_i + \mathbf{b}_s)^\mathsf{T} \mathbf{u}_s)}{\sum_i \exp(\tanh(\mathbf{W}_s \mathbf{h}_i + \mathbf{b}_s)^\mathsf{T} \mathbf{u}_s)},$$

$$\mathbf{v}_f = \sum_{i=1}^{n} \alpha_i \mathbf{h}_i,$$

where the meaning of $\mathbf{W}_s, \mathbf{b}_s$ and $\mathbf{u}_s$ are similar to that of $\mathbf{W}_w, \mathbf{b}_w$ and $\mathbf{u}_w$, respectively. Similarly, we can also obtain the encoded representations of other cases in the case triples, i.e., $\mathbf{v}^l_{f_{sim}}$, $\mathbf{v}^c_{f_{sim}}$ and $\mathbf{v}_{f_{dis}}$.

The Relational Attention Module. To refine the encoding process by extracting beneficial case relation information from case triples, we first calculate the attention vectors between case f and its similar and dissimilar cases in the representation space, as well as the anchor attention to itself,

$$\begin{aligned}
\mathbf{r}^l &= \mathbf{W}_l^3(\sigma(\mathbf{W}_l^1 \mathbf{v}_f + (\mathbf{W}_l^2 \mathbf{v}_f + \mathbf{b}_l^2))),\\
\mathbf{r}^l_{sim} &= \mathbf{W}_l^3(\sigma(\mathbf{W}_l^1 \mathbf{v}_f + (\mathbf{W}_l^2 \mathbf{v}^l_{f_{sim}} + \mathbf{b}_l^2))),\\
\mathbf{r}^l_{dis} &= \mathbf{W}_l^3(\sigma(\mathbf{W}_l^1 \mathbf{v}_f + (\mathbf{W}_l^2 \mathbf{v}_{f_{dis}} + \mathbf{b}_l^2))),\\
\mathbf{r}^c &= \mathbf{W}_c^3(\sigma(\mathbf{W}_c^1 \mathbf{v}_f + (\mathbf{W}_c^2 \mathbf{v}_f + \mathbf{b}_c^2))),\\
\mathbf{r}^c_{sim} &= \mathbf{W}_c^3(\sigma(\mathbf{W}_c^1 \mathbf{v}_f + (\mathbf{W}_c^2 \mathbf{v}^c_{f_{sim}} + \mathbf{b}_c^2))),\\
\mathbf{r}^c_{dis} &= \mathbf{W}_c^3(\sigma(\mathbf{W}_c^1 \mathbf{v}_f + (\mathbf{W}_c^2 \mathbf{v}_{f_{dis}} + \mathbf{b}_c^2))),
\end{aligned}$$

where $\mathbf{W}_l^1$, $\mathbf{W}_l^2$, $\mathbf{W}_l^3$ and $\mathbf{b}_l^2$ are weight matrices and bias vector for the first triple, the parameters for the second triple are similarly defined, and $\sigma(\cdot)$

[1] Note that the fact description encoder can be any existing encoder, and in the experiment section, we use a variety of encoders to verify the compatibility of CTM.

is the sigmoid activation function. Inspired by supervised contrastive learning (Schroff et al., 2015; Patro and Namboodiri, 2018), we impose a relational constraint on the two triples as an additional optimization objective,

$$\begin{aligned}\mathcal{L}_R = &\max(0, \beta_l + \|\mathbf{r}^l - \mathbf{r}^l_{sim}\|_2^2 - \|\mathbf{r}^l - \mathbf{r}^l_{dis}\|_2^2) \\ &+ \max(0, \beta_c + \|\mathbf{r}^c - \mathbf{r}^c_{sim}\|_2^2 - \|\mathbf{r}^c - \mathbf{r}^c_{dis}\|_2^2),\end{aligned} \tag{2}$$

where β_l and β_c are weight parameters. An intuitive explanation for the relational attention module is that in the attention vector between two cases, a higher attention value means that this dimension plays a greater role in the similarity of the two cases. By imposing the relational constraints in Eq.(2), we can further reduce noise from the attention vector between the current case and the corresponding similar case, which contributes to the similarity between the current case and dissimilar case.

The Category Decoder Module. To further avoid the influence between the high-frequency and the low-frequency cases, we set up a decoder for each of them to refine the decoding process. Since it is difficult for the model to know the category information of the current encoded representation in practice, we first impose a classification constraint to encourage the model to identify the category information more accurately,

$$\mathcal{L}_C = \mathcal{L}\left(\hat{y}_a, y_a\right), \tag{3}$$

where $\hat{y}_a = softmax(\mathbf{W}^2_{ca} * relu(\mathbf{W}^1_{ca}\mathbf{v}_f) + \mathbf{b}^2_{ca})$, $\mathbf{W}^1_{ca}$, $\mathbf{W}^2_{ca}$ and $\mathbf{b}^2_{ca}$ are weight matrices and bias vector. After obtaining the category label of the current case, we select the corresponding branch to decode the encoded representation. Note that the decoders on both branches have the same structure. Next, we use a unidirectional topological dependency structure similar to TopJudge (Zhong et al., 2018) as an example decoder[2]. The decoding process can be described as follows,

$$\begin{aligned}\begin{bmatrix}\mathbf{h}_l\\ \mathbf{c}_l\end{bmatrix} &= LSTMCell\left(\mathbf{v}_f, \begin{bmatrix}\overline{\mathbf{h}}_l\\ \overline{\mathbf{c}}_l\end{bmatrix}\right),\\ \begin{bmatrix}\overline{\mathbf{h}}_c\\ \overline{\mathbf{c}}_c\end{bmatrix} &= \left(\mathbf{W}_{c,l}\begin{bmatrix}\mathbf{h}_l\\ \mathbf{c}_l\end{bmatrix}\right) + \mathbf{b}_{c,l},\\ \begin{bmatrix}\mathbf{h}_c\\ \mathbf{c}_c\end{bmatrix} &= LSTMCell\left(\mathbf{v}_f, \begin{bmatrix}\overline{\mathbf{h}}_c\\ \overline{\mathbf{c}}_c\end{bmatrix}\right),\end{aligned}$$

where $\overline{\mathbf{h}}_l$ and $\overline{\mathbf{c}}_l$ are the initial hidden state and memory cell of the law article prediction task, $\mathbf{W}_{c,l}$ and $\mathbf{b}_{c,l}$ are the transformation matrix and bias vector that convert the task to charge prediction, and $\mathbf{h}_l$ and $\mathbf{h}_c$ are the decoded representations for these two tasks.

The Judgment Prediction Module. After obtaining the decoded representation of the current case, we use a fully connected layer to obtain the prediction of two different sub-tasks and the loss of multi-task prediction,

$$\begin{aligned}\hat{y}_l &= softmax(\mathbf{W}^l_p\mathbf{h}_l + \mathbf{b}^l_p),\\ \hat{y}_c &= softmax(\mathbf{W}^c_p\mathbf{h}_c + \mathbf{b}^c_p),\\ \mathcal{L}_M &= \mathcal{L}(\hat{y}_l, y_l) + \mathcal{L}(\hat{y}_c, y_c),\end{aligned} \tag{4}$$

where $\mathbf{W}^l_p$, $\mathbf{b}^l_p$ and $\mathbf{W}^c_p$, $\mathbf{b}^c_p$ are the parameters of the respective prediction tasks.

4 Experiments

In this section, we first introduce the experimental setup, and then conduct extensive empirical studies and show the effectiveness of our CTM.

4.1 Experiment Setup

Datasets. We use the two most common benchmark datasets in our experiments, i.e., CAIL-small and CAIL-big (Xiao et al., 2018). Following the settings of most previous works, we remove the cases with fewer than 10 meaningful words, and do not consider cases associated with multiple law articles or charges (Yang et al., 2019; Xu et al., 2020; Yue et al., 2021). Note that for a more comprehensive evaluation, we do not additionally remove the cases that contain law articles or charges with a frequency of lower than 100 as they do. Also, since CAIL-big does not provide a validation set, we divide the original training set for training and verification at a ratio of 9:1. The statistics of the datasets are shown in Table 1.

Table 1: Statistics of the datasets, i.e., CAIL-small and CAIL-big, used in the experiments.

	CAIL-small	CAIL-big
Training Cases	105,059	1,432,826
Validation Cases	14,266	159,372
Test Cases	27,953	186,523
Law Articles	177	181
Charges	191	193

Implementation Details. The baselines considered in the experiments include three existing representative methods, i.e., MTL (Zhong et al., 2018),

[2] Note that in the experiment section, we use decoders with other dependencies to verify the compatibility of CTM.

TopJudge (Zhong et al., 2018), and MPBFN (Yang et al., 2019), and two recent state-of-the-art methods, i.e., LADAN (Xu et al., 2020) and NeurJudge (Yue et al., 2021), where LADAN can be integrated with the first three methods to obtain three variants. All baselines are implemented on TensorFlow 1.15[3], Keras 2.3.1[4] or PyTorch 1.9.1[5] by referring to the source code and parameter settings provided in (Xu et al., 2020; Yue et al., 2021)[6,7]. We use four metrics for performance evaluation, including accuracy (Acc.), macro-recall (MR), macro-precision (MP) and macro-F1 (F1).

After some preliminary experiments, we fix the values of some additional parameters of CTM to reduce the search space, i.e., ϕ, β_l and β_c are set to 60%, 0.5 and 0.3, respectively. For all the methods, we set the maximum number of iterations to 20, and search the best batch size from $\{32, 64, 128\}$ by evaluating the accuracy of the law article prediction on the validation set. We also adopt an early stopping mechanism with a patience of 5 to avoid overfitting to the training set. By setting a random seed from 0 to 7, we run each method for eight times on Intel(R) Xeon(R) E5-2698 with 8 Tesla V100 GPU and report their average results[8].

4.2 Overall Results

If not specified, we use hierarchical Bi-GRU as the default encoder for reporting results, and constrain the fact description of a case to contain up to 15 sentences, where each sentence contains up to 100 words (Yang et al., 2019; Xu et al., 2020). The comparison results between our CTM and the baselines are shown in Table 2. We can see that our CTM consistently outperforms all the baselines on all the metrics across the two datasets of CAIL-small and CAIL-big. Furthermore, by comparing the results of F1, we find that considering more complex decoding dependency structure (i.e., MPBFN) is more prone to misclassification of low-frequency cases, and LADAN and NeurJudge alleviate this problem to some extent by refining the encoding process. Unlike them, our CTM can significantly further improve the model performance by introducing the case triples and customized modules.

4.3 Compatibility Analysis

As described in Sec. 3, since our CTM does not depend on a specific encoder and decoder, it can be easily integrated with existing decision prediction methods. We first study the compatibility of our CTM under different encoder choices. In addition to the default hierarchical Bi-GRU, we consider two common encoder choices, i.e., TextCNN (Kim, 2014) and Lawformer (Xiao et al., 2021). For TextCNN, we set the size of each filter to 64 and the filter widths to $(2, 3, 4, 5)$. Since Lawformer is a pre-trained language model with Longformer (Beltagy et al., 2020) for legal long documents, we directly use their provided model[9] for fine-tuning. We compare our CTM variants with different encoders against their respective baselines, i.e., adding the same decoder as our CTM for different encoders. We report the results on our CAIL-small in Table 3, from which we can see that our CTM brings significant improvement in all cases.

Next, we explore the compatibility of our CTM on different decoding structures. In addition to the default unidirectional topological dependency similar to TopJudge, we consider two decoding structures, i.e., ignoring the intra-task dependency similar to MTL and the bidirectional topological dependency similar to MPBFN. We compare our CTM variants with different decoding structures against their respective baselines, i.e., prepending the same encoder as our CTM for different decoding structures. The results on CAIL-small are shown in Table 4, from which we can see that our CTM has a significant advantage in all cases.

4.4 Ablation Studies

Moreover, we conduct ablation studies of our CTM to analyze the role played by each proposed new module. Specifically, we first consider the removal of the category decoder module (denoted as ‘w/o CD’), then consider using only the law article-based triple in the case triple module and relational attention module (denoted as ‘w/o CD+DS’), and finally remove these two modules (denoted as ‘w/o CD+CT+RA’). The results are shown in Table 5. We have the following observation: 1) By comparing ‘w/o CD+DS’ and ‘w/o CD+CT+RA’, the introduction of case triples is beneficial to the improvement of the model performance. 2) By comparing ‘w/o CD’ and w/o CD+DS’, multi-case triples are more efficient than single-case triples. 3) By com-

[3] https://www.tensorflow.org/

[4] https://keras.io/

[5] https://pytorch.org/

[6] https://github.com/prometheusXN/LADAN

[7] https://github.com/yuelinan/NeurJudge

[8] Note that the source codes are available at `https://github.com/dgliu/COLING22_CTM`

[9] https://huggingface.co/xcjthu/Lawformer

Table 2: Comparison results between our CTM and the baselines, where the significantly best results ($p \leq 0.05$ via two sample t-test) are marked in bold. Note that the accuracy of law article prediction is the main evaluation metric.

Datasets	CAIL-small								CAIL-big							
Tasks	Law Articles (%)				Charges (%)				Law Articles (%)				Charges (%)			
Metrics	Acc.	MR	MP	F1	Acc.	MR	MP	F1	Acc.	MR	MP	F1	Acc.	MR	MP	F1
MTL	77.06	60.76	63.21	59.60	81.72	68.31	71.54	67.57	95.68	61.79	73.47	64.92	95.53	68.53	80.97	71.79
TopJudge	77.35	60.73	62.94	59.63	81.54	68.09	70.22	67.27	95.73	61.78	73.84	64.99	95.53	67.55	80.06	70.90
MPBFN	72.77	50.55	53.25	48.74	75.41	56.15	59.28	55.15	94.13	48.83	60.99	51.26	93.60	50.06	64.01	52.96
MTL-LADAN	77.95	62.62	64.91	61.25	82.84	71.01	73.74	70.24	95.98	64.41	76.01	67.63	95.86	71.15	82.78	74.57
TopJudge-LADAN	78.45	63.65	65.95	62.39	83.19	71.88	74.00	71.06	96.08	64.91	77.07	68.27	95.90	70.81	82.42	74.08
MPBFN-LADAN	75.49	56.26	59.54	55.04	78.75	63.25	66.26	62.46	95.16	54.44	66.31	56.93	94.64	56.09	70.93	59.18
NeurJudge	78.27	62.20	66.34	61.74	81.01	64.93	69.55	65.26	95.87	65.04	76.65	68.12	94.86	64.88	79.58	68.66
CTM	**81.10**	**69.42**	**68.37**	**66.59**	**87.03**	**77.85**	**76.61**	**75.64**	**96.57**	**74.08**	**77.55**	**74.46**	**96.41**	**79.81**	**83.23**	**80.34**

Table 3: Comparison results of our CTM variants with different encoders and their respective baselines.

Tasks	Law Articles (%)				Charges (%)			
Metrics	Acc.	MR	MP	F1	Acc.	MR	MP	F1
TextCNN	75.97	54.30	60.84	53.52	80.05	60.72	65.22	60.54
Text-CTM	**80.37**	**67.08**	**65.53**	**63.50**	**85.78**	**73.97**	**73.03**	**70.80**
BiGRU	77.35	60.73	62.94	59.63	81.54	68.09	70.22	67.27
BiGRU-CTM	**81.10**	**69.42**	**68.37**	**66.59**	**87.03**	**77.85**	**76.61**	**75.64**
Lawformer	81.94	73.77	72.68	71.46	87.24	81.58	81.26	79.80
Law-CTM	**84.12**	**74.63**	**76.56**	**73.83**	**89.82**	**81.79**	**83.40**	**81.05**

Table 4: Comparison results of our CTM variants with different decoding structures and their respective baselines.

Tasks	Law Articles (%)				Charges (%)			
Metrics	Acc.	MR	MP	F1	Acc.	MR	MP	F1
MTL	77.06	60.76	63.21	59.60	81.72	68.31	71.54	67.57
MTL-CTM	**80.85**	**69.76**	**68.22**	**66.72**	**86.94**	**78.23**	**77.18**	**75.83**
TopJudge	77.35	60.73	62.94	59.63	81.54	68.09	70.22	67.27
TopJudge-CTM	**81.10**	**69.42**	**68.37**	**66.59**	**87.03**	**77.85**	**76.61**	**75.64**
MPBFN	72.77	50.55	53.25	48.74	75.41	56.15	59.28	55.15
MPBFN-CTM	**78.72**	**62.36**	**61.71**	**59.51**	**82.86**	**70.74**	**70.33**	**68.42**

Table 5: Results of the ablation studies on CAIL-small.

Tasks	Law Articles (%)				Charges (%)			
Metrics	Acc.	MR	MP	F1	Acc.	MR	MP	F1
CTM	**81.10**	**69.42**	**68.37**	**66.59**	**87.03**	**77.85**	**76.61**	**75.64**
w/o CD	78.11	63.16	65.12	61.87	82.68	70.91	72.69	69.93
w/o CD+DS	77.77	62.90	64.59	61.66	82.60	70.66	72.63	69.90
w/o CD+CT+RA	77.35	60.73	62.94	59.63	81.54	68.09	70.22	67.27

Table 6: Average accuracies of our CTM variants and their respective baselines across different frequency groups.

Tasks	Law Articles (%)				Charges (%)			
Groups	H1	H2	L1	L2	H1	H2	L1	L2
MTL	89.30	81.06	53.73	10.86	88.53	83.60	68.75	44.86
MTL-CTM	**89.89**	**84.73**	**68.50**	**21.83**	**89.74**	**85.28**	**83.39**	**59.92**
TopJudge	89.01	78.60	43.04	6.26	89.16	80.47	60.13	27.63
TopJudge-CTM	**90.49**	**84.03**	**69.96**	**22.50**	**89.65**	**84.67**	**84.03**	**63.94**
MPBFN	85.94	76.78	40.82	3.64	**86.34**	78.46	58.08	24.62
MPBFN-CTM	**87.78**	**80.13**	**60.74**	**12.87**	86.17	**79.52**	**75.71**	**52.94**

paring CTM and 'w/o CD', the introduction of the category decoder module results in greater gains. This may be due to the fact that refining the encoding process alone is still limited by the biased decoder training, and it is more beneficial to the model by refining the encoding and decoding processes jointly. Overall, the three customized modules we propose are necessary and can cooperate to achieve significant performance improvement.

4.5 Analysis of Gain Sources

In order to have a deeper understanding of the source of the performance gain, we compare and analyze the accuracy of the three variants of our CTM and the baselines on law articles and charges with different frequencies. The results of this fine-grained evaluation on CAIL-small are shown in Figure 5, where the IDs on the horizontal axis are sorted in a descending order of frequency. In Table 6, we also report the average accuracies of the CTM variants and their respective baselines across four different frequency groups, i.e., the top 20% (H1), 20% to 40% (H2), 40% to 70% (L1) and the rest (L2) of the label frequencies. Combining the above results, we can find that the improvement of our CTM increases significantly with decreasing frequency, which verifies the effectiveness of the designed case triples, especially for the low-frequency cases.

4.6 Visualization of Case Representations

Finally, we analyze the source of performance gain from the perspective of model training, i.e., compare the case representations generated by the baselines and its improved version via our CTM. We take MPBFN and MPBFN-CTM as an example due to space limitation. Specifically, in the case sampling module, we have obtained the high-frequency and low-frequency subsets of the law articles and

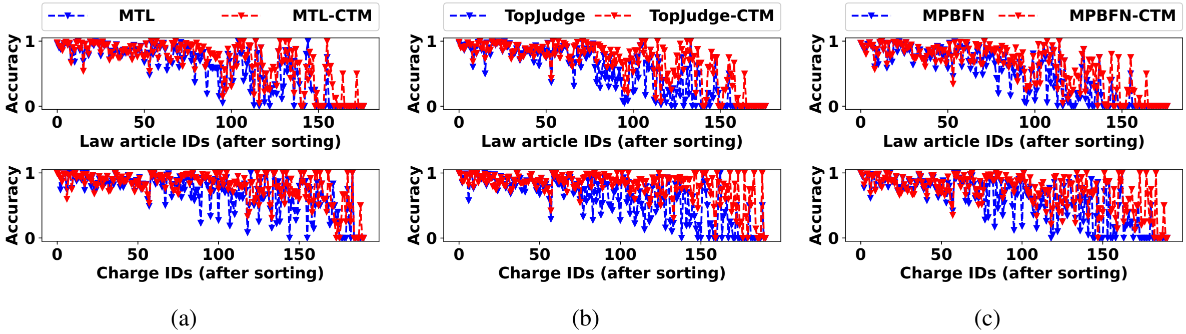
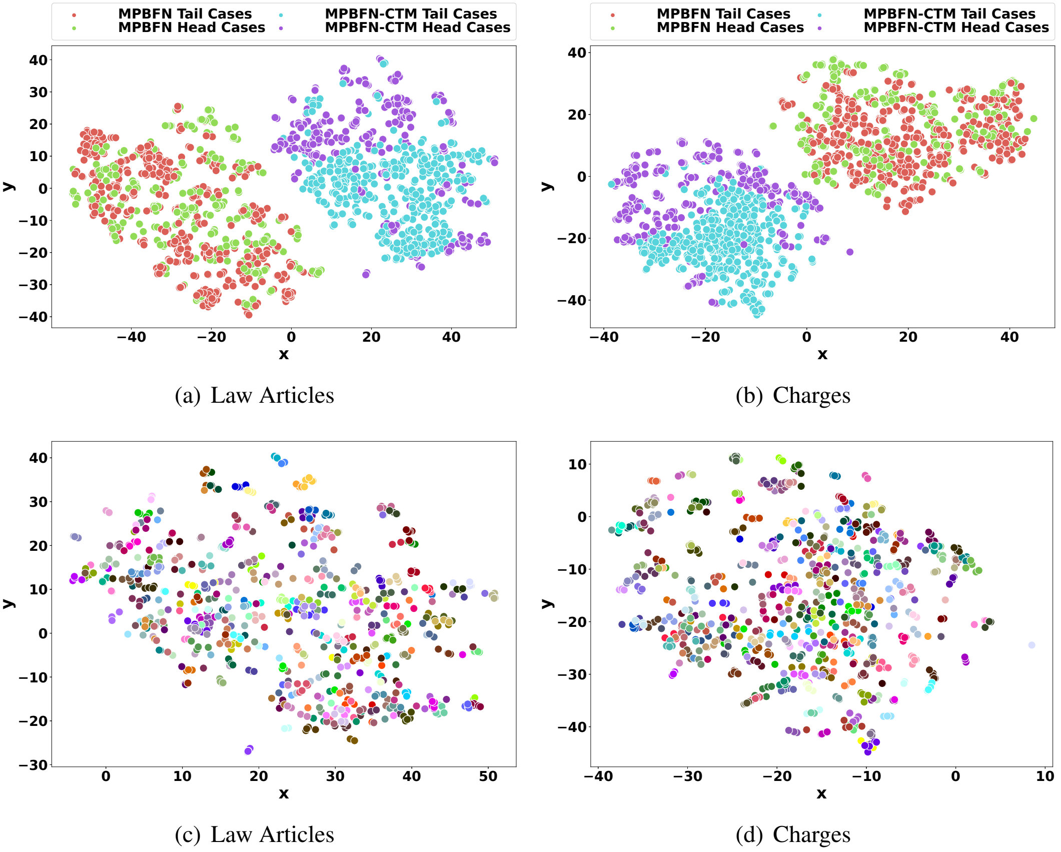


Figure 5: The prediction accuracy of our CTM variants and their respective baselines on law articles and charges with different frequencies from CAIL-small. Note that the IDs on the horizontal axis have been sorted in a descending order of frequency.



(a) Law Articles

(b) Charges

(c) Law Articles

(d) Charges

Figure 6: Visualization of the representations of some randomly sampled cases with high- and low-frequency law articles (a) and charges (b) on CAIL-small by MPBFN and MPBFN-CTM. The dots in (c) and (d) are fine-grained visualization of the representations obtained by MPBFN-CTM on each law article and charge, where the representations with the same law article or charge are clearly grouped.

charges. Then, we randomly sample 5 cases for each high-frequency (or low-frequency) law article and charge to construct their respective head (or tail) case sets. We respectively visualize the case representations generated by MPBFN and MPBFN-CTM on different sets.

The results are shown in Figure 6(a) and 6(b). We can find that the case representations generated by MPBFN have confusion on the head and tail sets (i.e., the green dots and red dots), and the case representations generated by MPBFN-CTM can cluster the head and tail sets separately and distinguish them effectively (i.e., the purple dots and blue dots). This clearly shows that the introduction of the case relations helps guide the encoder to learn the inter-class discrimination between the high-frequency and the low-frequency cases. We further present fine-grained visualization of the representations obtained by our CTM on each law article and each charge in Figure 6(c) and 6(d), respectively. As expected, we can see that most of the same law articles or charges, i.e., with the same colors, are clearly grouped.

5 Conclusions and Future Work

In this paper, we introduce some contrastive case relations to construct case triples as a new form of modeling, and propose a general judgment prediction framework with case triple modeling (CTM). Our CTM includes three new modules, i.e., a case sampling module for constructing case triples, a relational attention module for extracting information from case triples to refine the encoding process, and a category decoder module for refining the decoding process. Finally, we conduct extensive experiments on two public datasets and find that our CTM can effectively improve the performance of legal judgment prediction, especially for cases with low-frequency law articles or charges, and is also of good compatibility.

For future works, we plan to extend our CTM to more scenarios such as cases with multiple law articles or charges by further improving the corresponding case triple module and relational attention module. We are also interested in generalizing our CTM for prediction of the terms of penalty.

Acknowledgements

We thank the support of National Natural Science Foundation of China Nos. 61836005, 62272315 and 62172283.

References

Qiaoben Bao, Hongying Zan, Peiyuan Gong, Junyi Chen, and Yanghua Xiao. 2019. Charge prediction with legal attention. In *Proceedings of the 8th CCF International Conference on Natural Language Processing and Chinese Computing*, pages 447–458.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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.

Huajie Chen, Deng Cai, Wei Dai, Zehui Dai, and Yadong Ding. 2019a. Charge-based prison term prediction with deep gating network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 6362–6367.

Si Chen, Pengfei Wang, Wei Fang, Xingchen Deng, and Feng Zhang. 2019b. Learning to predict charges for judgment with legal graph. In *Proceedings of the 28th International Conference on Artificial Neural Networks*, pages 240–252.

Wenqing Chen, Jidong Tian, Liqiang Xiao, Hao He, and Yaohui Jin. 2020. Exploring logically dependent multi-task learning with causal inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2213–2225.

Yuh-Shyan Chen, Shin-Wei Chiang, and Meng-Luen Wu. 2021. A few-shot transfer learning approach using text-label embedding with legal attributes for law article prediction. *Applied Intelligence*, pages 1–19.

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.

Congqing He, Li Peng, Yuquan Le, Jiawei He, and Xiangyu Zhu. 2019. SECaps: A sequence enhanced capsule model for charge prediction. In *Proceedings of the 28th International Conference on Artificial Neural Networks*, pages 227–239.

Zhilong Hong, Qifei Zhou, Rong Zhang, Weiping Li, and Tong Mo. 2020. Legal feature enhanced semantic matching network for similar case matching. In *Proceedings of the 2020 International Joint Conference on Neural Networks*, pages 1–8.

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.

Xin Jiang, Hai Ye, Zhunchen Luo, Wenhan Chao, and Wenjia Ma. 2018. Interpretable rationale augmented charge prediction system. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 146–151.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Yuquan Le, Congqing He, Meng Chen, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Learning to predict charges for legal judgment via self-attentive capsule network. In *Proceedings of the 24th European Conference of Artificial Intelligence*, pages 1802–1809.

Shangbang Long, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2019. Automatic judgment prediction via legal reading comprehension. In *China National Conference on Chinese Computational Linguistics*, pages 558–572.

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*, pages 2727–2736.

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.

Malte Ostendorff, Elliott Ash, Terry Ruas, Bela Gipp, Julian Moreno-Schneider, and Georg Rehm. 2021. Evaluating document representations for content-based legal literature recommendations. In *Proceedings of the 18th International Conference on Artificial Intelligence and Law*, pages 109–118.

Badri Patro and Vinay P Namboodiri. 2018. Differential attention for visual question answering. In *Proceedings of the 31st IEEE Conference on Computer Vision and Pattern Recognition*, pages 7680–7688.

Dunlu Peng, Jiyin Yang, and Jing Lu. 2020. Similar case matching with explicit knowledge-enhanced text representation. *Applied Soft Computing*, 95:106514.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823.

Li Tang and Simon Clematide. 2021. Searching for legal documents at paragraph level: Automating label generation and use of an extended attention mask for boosting neural models of semantic similarity. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 114–122.

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

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. CAIL2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Tianyang Zhang, Xianpei Han, Zhen Hu, Heng Wang, et al. 2019. CAIL2019-SCM: A dataset of similar case matching in legal domain. *arXiv preprint arXiv:1911.08962*.

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.

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.

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. 2020. 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.