Distinguish Confusing Law Articles for Legal Judgment Prediction

Nuo Xu[1], **Pinghui Wang**[2,1]*, **Long Chen**[1], **Li Pan**[3], **Xiaoyan Wang**[4], **Junzhou Zhao**[1]*

[1]MOE NEKEY Lab, Xi'an Jiaotong University, China
[2]Shenzhen Research School, Xi'an Jiaotong University, China
[3]School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University
[4]Information Technology Service Center, The Supreme People's Court, China
`nxu@sei.xjtu.edu.cn, phwang@mail.xjtu.edu.cn,`
`chenlongche@stu.edu.cn, panli@sjtu.edu.cn,`
`wangxiaoyan@court.gov.cn, junzhouzhao@gmail.com`

Abstract

Legal Judgment Prediction (LJP) is the task of automatically predicting a law case's judgment results given a text describing its facts, which has excellent prospects in judicial assistance systems and convenient services for the public. In practice, confusing charges are frequent, because law cases applicable to similar law articles are easily misjudged. For addressing this issue, the existing method relies heavily on domain experts, which hinders its application in different law systems. In this paper, we present an end-to-end model, *LADAN*, to solve the task of LJP. To distinguish confusing charges, we propose a novel graph neural network to automatically learn subtle differences between confusing law articles and design a novel attention mechanism that fully exploits the learned differences to extract compelling discriminative features from fact descriptions attentively. Experiments conducted on real-world datasets demonstrate the superiority of our LADAN.

1 Introduction

Exploiting artificial intelligence techniques to assist legal judgment has become popular in recent years. Legal judgment prediction (**LJP**) aims to predict a case's judgment results, such as applicable law articles, charges, and terms of penalty, based on its fact description, as illustrated in Figure 1. LJP can assist judiciary workers in processing cases and offer legal consultancy services to the public. In the literature, LJP is usually formulated as a text classification problem, and several rule-based methods (Liu et al., 2004; Lin et al., 2012) and neural-based methods (Hu et al., 2018; Luo et al., 2017; Zhong et al., 2018) have been proposed.

The main drawback of existing methods is that they cannot solve the *confusing charges issue*.
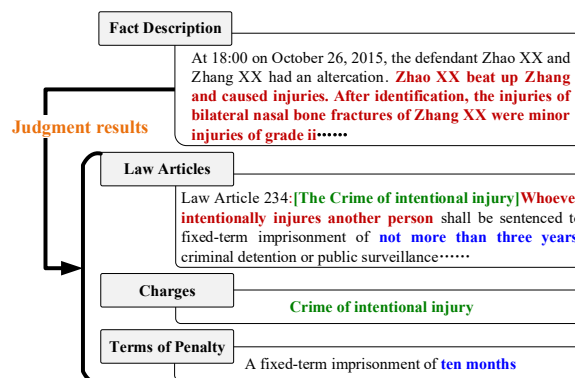
---
*Corresponding authors.

Figure 1: An illustration of the LJP. Generally, a judge needs to conduct professional analysis and reasoning on the fact description of the case, and then choose reasonable law articles, charges and the term of penalty to convict the offender.

That is, due to the high similarity of several law articles, their corresponding law cases can be easily misjudged. For example, in Figure 2, both *Article 385* and *Article 163* describe offenses of accepting bribes, and their subtle difference is whether the guilty parties are state staffs or not. The key to solving the confusing charges issue is how to capture essential but rare features for distinguishing confusing law articles. Hu et al. (2018) defined ten discriminative attributes to distinguish confusing charges. However, their method relies too much on experts to hinder its applications in a large number of laws. In practice, we desire a method that can automatically extract textual features from law articles to assist JLP. The most relevant existing work to this requirement is (Luo et al., 2017), which used an attention mechanism to extract features from fact descriptions with respect to a specific law article. As shown in Figure 3a, for each law article, an attention vector is computed, which is used to extract features from the fact description of a law case to predict whether the law article is applicable to the case. Nevertheless, the weakness

Article 385: The Crime of acceptance of bribes
Any state staffs who, taking advantage of his position, demands money or property from another person, or illegally accepts another person's money or property in return for securing benefits for the person *shall be guilty of acceptance of bribes.*

Article 163: Bribery crime of non-state staffs
The employees of companies, enterprises or other units who, taking advantage of his position, demands money or property from another person, or illegally accepts another person's money or property in return for securing benefits for the person *shall be guilty of bribery crime of non-state staffs.*

Article 389: Crime of offering bribes
Whoever, in order to seek illegitimate benefits, gives *any state staffs* with money and property, *shall be the crime of bribery*

Article 164: The crime of offering bribes to non-state staff
Whoever, in order to seek illegitimate benefits, gives employees of companies, enterprises or other units with money or property , *shall be guilty of bribing non-state staffs.*

Figure 2: Examples of confusing charges.

Figure 3: **a.** The fact-law attention model in (Luo et al., 2017). **b.** Our framework. *Variables α and β represent the encoded vectors learned for attentively extracting features from fact descriptions.*

is that they learn each law article's attention vector independently, and this may result in that similar attention vectors are learned for semantically close law articles; hence, it is ineffective in distinguishing confusing charges.

To solve the confusing charges issue, we propose an end-to-end framework, i.e., Law Article Distillation based Attention Network (LADAN). LADAN uses the difference among similar law articles to attentively extract features from law cases' fact descriptions, which is more effective in distinguishing confusing law articles, and improve the performance of LJP. To obtain the difference among similar law articles, a straightforward way is to remove duplicated texts between two law articles and only use the leftover texts for the attention mechanism. However, we find that this method may generate the same leftover texts for different law article, and generate misleading information to LJP. As shown in Fig. 2, if we remove the duplicated phrases and sentences between Article 163 and Article 385 (i.e., the red text in Fig. 2), and between Article 164 and Article 389 (i.e., the pink text in Fig. 2), respectively, then Article 385 and Article 389 will be almost same to each other (i.e., the blue text in Fig. 2).

We design LADAN based on the following observation: it is usually easy to distinguish dissimilar law articles as sufficient distinctions exist, but challenging to discriminate similar law articles due to the few useful features. We first group law articles into different communities, and law articles in the same community are highly similar to each other. Then we propose a graph-based representation learning method to automatically explore the difference among law articles and com-
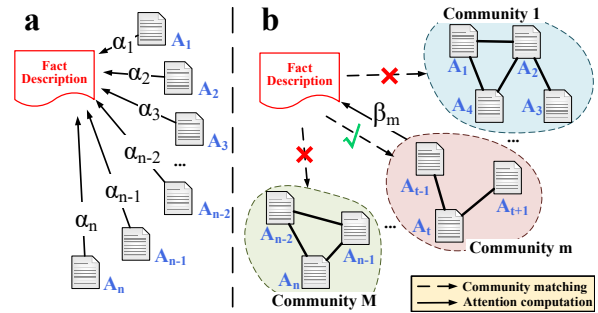
pute an attention vector for each community. For an input law case, we learn both macro- and micro-level features. Macro-level features are used for predicting which community includes the applicable law articles. Micro-level features are attentively extracted by the attention vector of the selected community for distinguishing confusing law articles within the same community. Our main contributions are summarized as follows:

(1) We develop an end-to-end framework, i.e., LADAN, to solve the LJP task. It addresses the confusing charges issue by mining similarities between fact descriptions and law articles as well as the distinctions between confusing law articles.

(2) We propose a novel graph distillation operator (GDO) to extract discriminative features for effectively distinguishing confusing law articles.

(3) We conduct extensive experiments on real-world datasets. The results show that our model outperforms all state-of-the-art methods.

2 Related Work

Our work solves the problem of the confusing charge in the LJP task by referring to the calculation principle of graph neural network (GNN). Therefore, in this section, we will introduce related works from these two aspects.

2.1 Legal Judgment Prediction

Existing approaches for legal judgment prediction (LJP) are mainly divided into three categories. In early times, works usually focus on analyzing existing legal cases in specific scenarios with mathematical and statistical algorithms (Kort, 1957; Nagel, 1963; Keown, 1980; Lauderdale and Clark, 2012). However, these methods are limited to small datasets with few labels. Later, a number of

machine learning-based methods (Lin et al., 2012; Liu et al., 2004; Sulea et al., 2017) were developed to solve the problem of LJP, which almost combine some manually designed features with a linear classifier to improve the performance of case classification. The shortcoming is that these methods rely heavily on manual features, which suffer from the generalization problem.

In recent years, researchers tend to exploit neural networks to solve LJP tasks. Luo et al. (2017) propose a hierarchical attentional network to capture the relation between fact description and relevant law articles to improve the charge prediction. Zhong et al. (2018) model the explicit dependencies among subtasks with scalable directed acyclic graph forms and propose a topological multi-task learning framework for effectively solving these subtasks together. Yang et al. (2019) further refine this framework by adding backward dependencies between the prediction results of subtasks. To the best of our knowledge, Hu et al. (2018) are the first to study the problem of discriminating confusing charges for automatically predicting applicable charges. They manually define 10 discriminative attributes and propose to enhance the representation of the case fact description by learning these attributes. This method relies too much on experts and cannot be easily extended to different law systems. To solve this issue, we propose a novel attention framework that automatically extracts differences between similar law articles to enhance the representation of fact description.

2.2 Graph Neural Network

Due to its excellent performance in graph structure data, GNN has attracted significant attention (Kipf and Welling, 2017; Hamilton et al., 2017; Bonner et al., 2019). In general, existing GNNs focus on proposing different aggregation schemes to fuse features from the neighborhood of each node in the graph for extracting richer and more comprehensive information: Kipf et al. (2017) propose graph convolution networks which use mean pooling to pool neighborhood information; GraphSAGE (Hamilton et al., 2017) concatenates the node's features and applies mean/max/LSTM operators to pool neighborhood information for inductively learning node embeddings; MR-GNN (Xu et al., 2019) aggregates the multi-resolution features of each node to exploit node information, subgraph information, and global information together; Besides, Message Passing Neural Networks (Gilmer et al., 2017) further consider edge information when doing the aggregation. However, the aggregation schemes lead to the over-smoothing issue of graph neural networks (Li et al., 2018), i.e., the aggregated node representations would become indistinguishable, which is entirely contrary to our goal of extracting distinguishable information. So in this paper, we propose our distillation operation, based on a distillation strategy instead of aggregation schemes, to extract the distinguishable features between similar law articles.

3 Problem Formulation

In this section, we introduce some notations and terminologies, and then formulate the LJP task.

Law Cases. Each law case consists of a *fact description* and several *judgment results* (cf. Figure 1). The fact description is represented as a text document, denoted by f. The judgment results may include *applicable law articles*, *charges*, *terms of penalty*, etc. Assume there are t kinds of judgment results, and the i-th judgment result is represented as a categorical variable y_i which takes value from set Y_i. Then, a law case can be represented by a tuple $(f, y_1, \ldots, y_t)$.

Law Articles. Law cases are often analyzed and adjudicated according to a legislature's *statutory law* (also known as, *written law*). Formally, we denote the statutory law as a set of *law articles* $\mathcal{L} \triangleq \{L_1, \ldots, L_m\}$ where m is the number of law articles. Similar to the fact description of cases, we also represent each law article L_i as a document.

Legal Judgment Prediction. In this paper, we consider three kinds of judgment results: *applicable law articles*, *charges*, and *terms of penalty*. Given a training dataset $D \triangleq \{(f, y_1, y_2, y_3)_z\}_{z=1}^q$ of size q, we aim to train a model $F(\cdot)$ that can predict the judgment results for any test law case with a fact description f_{test}, i.e., $F(f_{test}, \mathcal{L}) = (\hat{y}_1, \hat{y}_2, \hat{y}_3)$, where $\hat{y}_i \in Y_i$, $i = 1, 2, 3$. Following (Zhong et al., 2018; Yang et al., 2019), we assume each case has only one applicable law article.
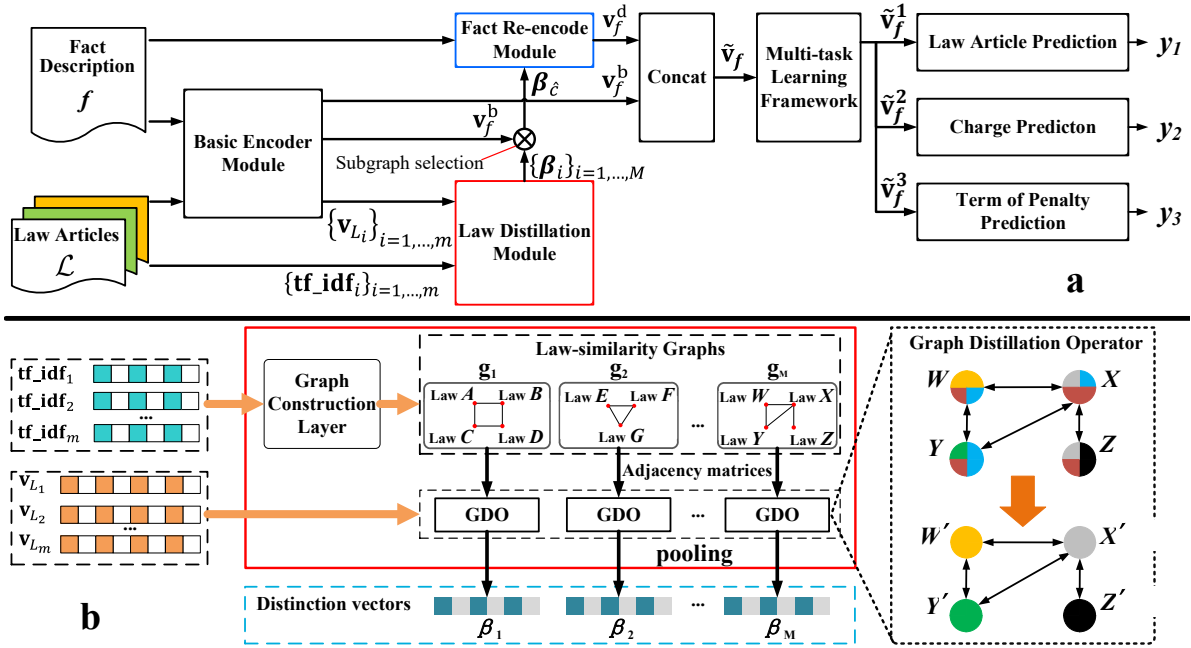
Figure 4: **a.** Overview of our framework *LADAN*: it takes the fact descriptions of cases and the text definitions of law articles as inputs, then extracts the basic representation $\mathbf{v}_f^b$ and distinguishing representation $\mathbf{v}_f^d$ of the fact descriptions through the basic encoder and the re-encoder, and finally combines this two representations for the downstream prediction tasks; **b.** Law Distillation Module: this module communizes law articles and distills the distinguishable features of each community for attention calculation of the re-encoder.

4 Our Method

4.1 Overview

In our framework LADAN (cf. Fig. 4a), the fact description of a case is represented by two parts: a *basic representation*, denoted by $\mathbf{v}_f^b$, and a *distinguishable representation*, denoted by $\mathbf{v}_f^d$. The basic representation $\mathbf{v}_f^b$ contains basic semantic information for matching a group of law articles that may apply to the case. In contrast, the distinguishable representation $\mathbf{v}_f^d$ captures features that can effectively distinguish confusing law articles. The concatenation of $\mathbf{v}_f^b$ and $\mathbf{v}_f^d$ is fed into subsequent classifiers to predict the labels of the JLP task.

As we mentioned, it is easy to distinguish dissimilar law articles as sufficient distinctions exist, and the difficulty in solving confusing charges lies in extracting distinguishable features of similar law articles. To obtain the basic representation $\mathbf{v}_f^b$, therefore, we use one of the popular document encoding methods (e.g., CNN encoder (Kim, 2014) and Bi-RNN encoder (Yang et al., 2016)). To learn the distinguishable representation $\mathbf{v}_f^d$, we use a *law distillation module* first to divide law articles to several communities to ensure that the law articles in each community are highly similar, and then extract each community i's distinction vector

(or, distinguishable features) β_i from the basic representation of law articles in community i. Given the case's fact description, from all communities' distinction vectors, we select the most relevant one (i.e., $\beta_{\hat{c}}$ in Fig. 4(a)) for attentively extracting the distinguishable features $\mathbf{v}_f^d$ in the *fact re-encode module*. In the follows, we elaborate law distillation module (Sec. 4.2) and fact re-encode module (Sec. 4.3) respectively.

4.2 Distilling Law Articles

A case might be misjudged due to the high similarity of some law articles. To alleviate this problem, we design a law distillation module (cf. Fig. 4 b) to extract distinguishable and representative information from all law articles. Specifically, it first uses a *graph construction layer* (**GCL**) to divide law articles into different communities. For each law article community, a graph distillation layer is applied to learn its discriminative representation, hereinafter, called *distinction vector*.

4.2.1 Graph Construction Layer

To find probably confusing law articles, we first construct a fully-connected graph G^* for all law articles $\mathcal{L}$, where the weight on the edge between a pair of law article $L_i, L_j \in \mathcal{L}$ is defined as

the cosine similarity between their TF-IDF (Term Frequency-Inverse Document Frequency) representations $\mathbf{tf\_idf}_i$ and $\mathbf{tf\_idf}_j$. Since confusing law articles are usually semantically similar and there exists sufficient information to distinguish dissimilar law articles, we remove the edges with weights less than a predefined threshold τ from graph G^*. By setting an appropriate τ, we obtain a new graph $G = \{g_i\}_{i=1}^{M}$ composed of several disconnected subgraphs $g_1, \ldots, g_M$ (or, communities), where each $g_i, i = 1, \ldots, M$ contains a specific community of probably confusing articles. Our later experimental results demonstrate that this easy-to-implement method effectively improves the performance of LADAN.

4.2.2 Graph Distillation Layer

To extract the distinguishable information from each community g_i, a straightforward way is to delete duplicate words and sentences presented in law articles within the community (as described in Sec. 1). In addition to introducing significant errors, this simple method cannot be plugged into end-to-end neural architectures due to its non-differentiability. To overcome the above issues, inspired by the popular graph convolution operator (**GCO**) (Kipf and Welling, 2017; Hamilton et al., 2017; Veličković et al., 2017), we propose a *graph distillation operator* (**GDO**) to effectively extract distinguishable features. Different from GCO, which computes the message propagation between neighbors and aggregate these messages to enrich representations of nodes in the graph, the basic idea behind our GDO is to learn effective features with distinction by removing similar features between nodes.

Specifically, for an arbitrary law article L_i, GDO uses a trainable weight matrix Ψ to capture similar information between it and its neighbors in graph G, and a matrix Φ to extract effective semantic features of L_i. At each layer $l \geq 0$, the aggregation of similar information between L_i and its neighbors is removed from its representation, that is,

$$\mathbf{v}_{L_i}^{(l+1)} = \Phi^{(l)} \mathbf{v}_{L_i}^{(l)} - \sum_{L_j \in N_i} \frac{\Psi^{(l)}[\mathbf{v}_{L_i}^{(l)}, \mathbf{v}_{L_j}^{(l)}]}{|N_i|} + \mathbf{b}^{(l)}$$

where $\mathbf{v}_{L_i}^{(l)} \in \mathbb{R}^{d_l}$ refers to the representation of law L_i in the lth graph distillation layer, N_i refers to the neighbor set of L_i in graph G, $\mathbf{b}^{(l)}$ is the

bias, and $\Phi^{(l)} \in \mathbb{R}^{d_{l+1} \times d_l}$ and $\Psi^{(l)} \in \mathbb{R}^{d_{l+1} \times 2d_l}$ are the trainable self weighted matrix and the neighbor similarity extracting matrix respectively. Note that d_l is the dimension of the feature vector in the lth graph distillation layer. We set $d_0 = d_s$, where d_s is the dimension of basic representations $\mathbf{v}_f^{\text{b}}$ and $\mathbf{v}_{L_i}$. Similar to GCO, our GDO also supports multi-layer stacking.

Using GDO with H layers, we output law article representation of the last layer, i.e., $\mathbf{v}_{L_i}^{(H)} \in \mathbb{R}^{d_H}$, which contains rich distinguishable features that can distinguish law article L_i from the articles within the same community. To further improve law articles' distinguishable features, for each subgraph $g_i, i = 1, 2, \ldots, M$ in graph G, we compute its distinction vector β_i by using pooling operators to aggregate the distinguishable features of articles in g_i. Formally, β_i is computed as:

$$\beta_i = [\text{MaP}(\{\mathbf{v}_{L_i}^{(H)}\}_{L_j \in g_i}), \text{MiP}(\{\mathbf{v}_{L_i}^{(H)}\}_{L_j \in g_i})]$$

where $\text{MaP}(\cdot)$ and $\text{MiP}(\cdot)$ are the element-wise max pooling and element-wise min pooling operators respectively.

4.3 Re-encoding Fact with Distinguishable Attention

To capture a law case's distinguishable features from its fact description f, we firstly define the following linear function, which is used to predict its most related community $g_{\hat{c}}$ in graph G:

$$\hat{\mathbf{X}} = \text{softmax}(\mathbf{W}_g \mathbf{v}_f^{\text{b}} + \mathbf{b}_g) \qquad (1)$$

where $\mathbf{v}_f^{\text{b}}$ is the basic representation of fact description f, $\mathbf{W}_g \in \mathbb{R}^{M \times d_s}$ and $\mathbf{b}_g \in \mathbb{R}^M$ are the trainable weight matrix and bias respectively. Each element $\hat{X}_i \in \hat{\mathbf{X}}$, $i = 1, ..., M$ reflects the closeness between fact description f and law articles community g_i. The most relevant community $g_{\hat{c}}$ is computed as

$$\hat{c} = \arg \max_{i=1,...,M} \hat{X}_i.$$

Then, we use the corresponding community's distinction vector $\beta_{\hat{c}}$ to attentively extract distinguishable features from fact description f.

Inspired by (Yang et al., 2016), we attentively extract distinguishable features based on word-level and sentence-level Bi-directional Gated Recurrent Units (Bi-GRUs). Specifically, for each input sentence $S_i = [w_{i,1}, \cdots, w_{i,n_i}]$ in fact description f, word-level Bi-GRUs will output a hidden

state sequence, that is,

$$\mathbf{h}_{i,j} = [\overrightarrow{\text{GRU}}(\mathbf{w}_{i,j}), \overleftarrow{\text{GRU}}(\mathbf{w}_{i,j})], \quad j = 1, ..., n_i,$$

where $\mathbf{w}_{i,j}$ represents the word embedding of word $w_{i,j}$ and $\mathbf{h}_{i,j} \in \mathbb{R}^{d_w}$. Based on this hidden state sequence and the distinction vector $\beta_{\hat{c}}$, we calculate an attentive vector $[\alpha_{i,1}, \ldots, \alpha_{i,n_i}]$, where each $\alpha_{i,j}$ evaluates the discrimination ability of word $w_{i,j} \in S_i$. $\alpha_{i,j}$ is formally computed as:

$$\alpha_{i,j} = \frac{\exp(\tanh(\mathbf{W}_w \mathbf{h}_{i,j})^{\mathsf{T}}(\mathbf{W}_{gw}\beta_{\hat{c}}))}{\sum_j \exp(\tanh(\mathbf{W}_w \mathbf{h}_{i,j})^{\mathsf{T}}(\mathbf{W}_{gw}\beta_{\hat{c}}))},$$

where $\mathbf{W}_w$ and $\mathbf{W}_{gw}$ are trainable weight matrices. Then, we get a representation of sentence S_i as:

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

where n_i denotes the word number in sentence S_i.

By the above word-level Bi-GRUs, we get a sentence representations sequence $[\mathbf{v}_{s_1}, \ldots, \mathbf{v}_{s_{n_f}}]$, where n_f refers to the number of sentences in the fact description f. Based on this sequence, similarly, we build sentence-level Bi-GRUs and calculate a sentence-level attentive vector $[\alpha_1, \ldots, \alpha_{n_f}]$ that reflects the discrimination ability of each sentence, and then get the fact's distinguishable representation $\mathbf{v}_f^{\mathrm{d}} \in \mathbb{R}^{d_s}$. Our sentence-level Bi-GRUs are formulated as:

$$\mathbf{h}_i = [\overrightarrow{\text{GRU}}(\mathbf{v}_{s_i}), \overleftarrow{\text{GRU}}(\mathbf{v}_{s_i})], \quad i = 1, 2, ..., n_f,$$

$$\alpha_i = \frac{\exp(\tanh(\mathbf{W}_\mathbf{s}\mathbf{h}_i)^{\mathsf{T}}(\mathbf{W}_{gs}\beta_{\hat{c}}))}{\sum_i \exp(\tanh(\mathbf{W}_\mathbf{s}\mathbf{h}_i)^{\mathsf{T}}(\mathbf{W}_{gs}\beta_{\hat{c}}))},$$

$$\mathbf{v}_f^{\mathrm{d}} = \sum_i \alpha_i \mathbf{h}_i.$$

4.4 Prediction and Training

We concatenate the basic representation $\mathbf{v}_f^{\mathrm{b}}$ and the distinguishable representation $\mathbf{v}_f^{\mathrm{d}}$ as the final representation of fact description f, i.e., $\tilde{\mathbf{v}}_f = [\mathbf{v}_f^{\mathrm{b}}, \mathbf{v}_f^{\mathrm{d}}]$. Based on $\tilde{\mathbf{v}}_f$, we generate a corresponding feature vector $\tilde{\mathbf{v}}_f^j$ for each subtask t_j, $j = 1, 2, 3$ mentioned in Sec. 3, i.e., t_1: **law article prediction**; t_2: **charge prediction**; t_3: **term of penalty prediction**. To obtain the prediction for each subtask, we use a linear classifier:

$$\hat{y}_j = \text{softmax}(\mathbf{W}_p^j \tilde{\mathbf{v}}_f^j + \mathbf{b}_p^j),$$

where $\mathbf{W}_p^j$ and $\mathbf{b}_p^j$ are parameters specific to task t_j. For training, we compute a cross-entropy loss function for each subtask and take the loss sum of all subtasks as the overall prediction loss:

$$\mathscr{L}_p = -\sum_{j=1}^{3} \sum_{k=1}^{|Y_j|} y_{j,k} \log(\hat{y}_{j,k}),$$

where $|Y_j|$ denotes the number of different classes (or, labels) for task t_j and $[y_{j,1}, y_{j,2}, \ldots, y_{j,|Y_j|}]$ refers to the ground-truth vector of task t_j. Besides, we also consider the loss of law article community prediction (i.e., Eq. 1):

$$\mathscr{L}_c = -\lambda \sum_{j=1}^{M} X_j \log(\hat{X}_j),$$

where $[X_1, X_2, \ldots, X_M]$ is the ground-truth vector of the community including the correct law article applied to the law case. In summary, our final overall loss function is:

$$\mathscr{L} = \mathscr{L}_p + \mathscr{L}_c \qquad (2)$$

5 Experiments

5.1 Datasets

To evaluate the performance of our method, we use the publicly available datasets of the **C**hinese **AI** and **L**aw challenge (CAIL2018)[1] (Xiao et al., 2018): *CAIL-small* (the exercise stage dataset) and *CAIL-big* (the first stage dataset). The case samples in both datasets contain fact description, applicable law articles, charges, and the terms of penalty. For data processing, we first filter out samples with fewer than 10 meaningful words. To be consistent with state-of-the-art methods, we filter out the case samples with multiple applicable law articles and multiple charges. Meanwhile, referring to (Zhong et al., 2018), we only keep the law articles and charges that apply to not less than 100 corresponding case samples and divide the terms of penalty into non-overlapping intervals. The detailed statistics of the datasets are shown in Table 1.

5.2 Baselines and Settings

Baselines. We compare LADAN with some baselines, including:

---

[1] http://cail.cipsc.org.cn/index.html

Dataset	CAIL-small	CAIL-big
#Training Set Cases	101,619	1,587,979
#Test Set Cases	26,749	185,120
#Law Articles	103	118
#Charges	119	130
#Term of Penalty	11	11

Table 1: Statistics of datasets.

- **CNN (Kim, 2014):** a CNN-based model with multiple filter window widths for text classification.

- **HARNN (Yang et al., 2016):** an RNN-based neural network with a hierarchical attention mechanism for document classification.

- **FLA (Luo et al., 2017):** a charge prediction method that uses an attention mechanism to capture the interaction between fact description and applicable laws.

- **Few-Shot (Hu et al., 2018):** a discriminating confusing charge method, which extracts features about ten predefined attributes from fact descriptions to enforce semantic information.

- **TOPJUDGE (Zhong et al., 2018):** a topological multi-task learning framework for LJP, which formalizes the explicit dependencies over subtasks in a directed acyclic graph.

- **MPBFN-WCA (Yang et al., 2019):** a multi-task learning framework for LJP with multi-perspective forward prediction and backward verification, which is the state-of-the-art method.

Similar to existing works (Luo et al., 2017; Zhong et al., 2018), we train the baselines CNN, HLSTM and FLA using a multi-task framework (recorded as MTL) and select a set of the best experimental parameters according to the range of the parameters given in their original papers. Besides, we use our method LADAN with the same multi-task framework (i.e., Landan+MTL, LADAN+TOPJUDGE, and LADAN+MPBFN) to demonstrate our superiority in feature extraction.

Experimental Settings. We use the THU-LAC (Sun et al., 2016) tool to get the word segmentation because all case samples are in Chinese. Afterward, we use the Skip-Gram model (Mikolov et al., 2013) to pre-train word embeddings on these case documents, where the model's embedding size and frequency threshold are set to 200 and 25 respectively. Meanwhile, we set the maximum document length as 512 words for CNN-based models in baselines and set the maximum sentence length to 100 words and maximum document length to 15 sentences for LSTM-based models. As for hyperparameters setting, we set the dimension of all latent states (i.e., d_w, d_s, d_l and d_f) as 256 and the threshold τ as 0.3. In our method LADAN, we use two graph distillation layers, and a Bi-GRU with a randomly initialized attention vector u is adopted as the basic document encoder. For training, we set the learning rate of Adam optimizer to 10^{-3}, and the batch size to 128. After training every model for 16 epochs, we choose the best model on the validation set for testing.[2]

5.3 Experimental Results

To compare the performance of the baselines and our methods, we choose four metrics that are widely used for multi-classification tasks, including accuracy (Acc.), macro-precision (MP), macro-recall (MR), and macro-F1 (F1). Since the problem of confusing charges often occurs between a few categories, the main metric is the F1 score. Tables 2 and 3 show the experimental results on datasets CAIL-small and CAIL-big, respectively. Our method LADAN performs the best in terms of all evaluation metrics. Because both CAIL-small and CAIL-big are imbalanced datasets, we focus on comparing the F1-score, which more objectively reflects the effectiveness of our LADAN and other baselines. Compared with the state-of-the-art MPBFN-WCA, LADAN improved the F1-scores of law article prediction, charge prediction, and term of penalty prediction on dataset CAIL-small by 2.02%, 2.42% and 4.20% respectively, and about 3.18%, 1.44% and 5.79% on dataset CAIL-big. Meanwhile, the comparison under the same multi-task framework (i.e., MTL, TOPJUDGE, and MPBFN) shows that our LADAN extracted more effective features from fact descriptions than all baselines. Meanwhile, we can observe that the performance of Few-shot on charge prediction is close to LADAN, but its performance on the term of penalty prediction is far from ideal. It is because the ten predefined attributes of Few-Shot are only effective for identifying charges, which also proves the robustness

---

[2]Our source codes are available at `https://github.com/prometheusXN/LADAN`

Tasks	Law Articles				Charges				Term of Penalty			
Metrics	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
FLA+MTL	77.74	75.32	74.36	72.93	80.90	79.25	77.61	76.94	36.48	30.94	28.40	28.00
CNN+MTL	78.71	76.02	74.87	73.79	82.41	81.51	79.34	79.61	35.40	33.07	29.26	29.86
HARNN+MTL	79.79	75.26	76.79	74.90	83.80	82.44	82.78	82.12	36.17	34.66	31.26	31.40
Few-Shot+MTL	79.30	77.80	77.59	76.09	83.65	80.84	82.01	81.55	36.52	35.07	26.88	27.14
TOPJUDGE	79.88	79.77	73.67	73.60	82.10	83.60	78.42	79.05	36.29	34.73	32.73	29.43
MPBFN-WCA	79.12	76.30	76.02	74.78	82.14	82.28	80.72	80.72	36.02	31.94	28.60	29.85
LADAN+MTL	**81.20**	**78.24**	**77.38**	**76.47**	**85.07**	**83.42**	**82.52**	**82.74**	**38.29**	**36.16**	**32.49**	**32.65**
LADAN+TOPJUDGE	**81.53**	**78.62**	**78.29**	**77.10**	**85.12**	**83.64**	**83.57**	**83.14**	**38.34**	**36.39**	**32.75**	**33.53**
LADAN+MPBFN	**82.34**	**78.79**	**77.59**	**76.80**	**84.83**	**83.33**	**82.80**	**82.85**	**39.35**	**36.94**	**33.25**	**34.05**

Table 2: Judgment prediction results on CAIL-small.

Tasks	Law Articles				Charges				Term of Penalty			
Metrics	Acc.	MP	MR	F1	Acc.	MP	MR	F1	Acc.	MP	MR	F1
FLA+MTL	93.23	72.78	64.30	66.56	92.76	76.35	68.48	70.74	57.63	48.93	45.00	46.54
CNN+MTL	95.84	83.20	75.31	77.47	95.74	86.49	79.00	81.37	55.43	45.13	38.85	39.89
HARNN+MTL	95.63	81.48	74.57	77.13	95.58	85.59	79.55	81.88	57.38	43.50	40.79	42.00
Few-Shot+MTL	96.12	85.43	80.07	81.49	96.04	88.30	80.46	83.88	57.84	47.27	42.55	43.44
TOPJUDGE	95.85	84.84	74.53	77.50	95.78	86.46	78.51	81.33	57.34	47.32	42.77	44.05
MPBFN-WCA	96.06	85.25	74.82	78.36	95.98	89.16	79.73	83.20	58.14	45.86	39.07	41.39
LADAN+MTL	**96.57**	**86.22**	**80.78**	**82.36**	**96.45**	**88.51**	**83.73**	**85.35**	**59.66**	**51.78**	**45.34**	**46.93**
LADAN+TOPJUDGE	**96.62**	**86.53**	**79.08**	**81.54**	**96.39**	**88.49**	**82.28**	**84.64**	**59.70**	**51.06**	**45.46**	**46.96**
LADAN+MPBFN	**96.60**	**86.42**	**80.37**	**81.98**	**96.42**	**88.45**	**83.08**	**84.95**	**59.85**	**51.75**	**45.59**	**47.18**

Table 3: Judgment prediction results on CAIL-big.

of our LADAN. The highest MP- and MR-scores of LADAN also demonstrates its ability to distinguish confusing law articles. Note that all methods' performance on dataset CAIL-big is better than that on CAIL-small, which is because the training set on CAIL-big is more adequate.

5.4 Ablation Experiments

To further illustrate the significance of considering the difference between law articles, we conducted ablation experiments on model *LADAN+MTL* with dataset CAIL-small. To prove the effectiveness of our graph construction layer (**GCL**), we build a LADAN model with the GCL's removing threshold $\tau = 0$ (i.e., "-no GCL" in Table 4), which directly applies the GDO on the fully-connected graph G^* to generate a global distinction vector β_g for re-encoding the fact description. To verify the effectiveness of our graph distillation operator (**GDO**), we build a no-GDO LADAN model (i.e., "-no GDO" in Table 4), which directly pools each subgraph g_i to a distinction vector β_i without GDOs. To evaluate the importance of considering the difference among law articles, we remove both GCL and GDO from LADAN by setting $\tau = 1.0$ (i.e., "-no both" in Table 4), i.e., each law article independently extracts the attentive feature from fact description. In Table 4, we

Tasks	Law		Charge		Penalty	
Metrics	Acc.	F1	Acc.	F1	Acc.	F1
LADAN+MTL	81.20	76.47	85.07	83.14	38.29	32.65
-no GCL	80.46	75.98	84.04	82.33	37.80	31.85
-no GDO	80.82	76.19	84.65	82.50	36.69	31.62
-no both	79.79	74.97	83.72	82.02	34.87	31.34

Table 4: Ablation analysis on CAIL-small.

see that both GCL and GDO effectively improve the performance of LADAN. GCL is more critical than GDO because GDO has a limited performance when the law article communities obtained by GCL are not accurate. When removing both GCL and GDO, the accuracy of LADAN decreases to that of HARNN+MTL, which powerfully demonstrates the effectiveness of our method exploiting differences among similar law articles.

5.5 Case Study

To intuitively verify that LADAN effectively extracts distinguishable features, we visualize the attention of LADAN's encoders. Figure 5 shows two law case examples, each for *Article 385* and *Article 163*, respectively, where the darker the word is, the higher the attention weight it gets in the corresponding encoder, i.e., its information is more important to the encoder. For the basic encoder, we see that the vital information in these two cases is very similar, which both contain the

Basic Encoder:

In the second half of 2017, the defendant Tian, while working in the Cadastral Unit of Luocheng Branch of Luohe City Land and Resources Bureau, used his position responsibility for cadastral registration to provide Zhao with a blank cadastral registration form, forged a false cadastral file, and accepted Zhao a certain amount of cash of 25,000 yuan for personal consumption.

Fact Re-encoder:

In the second half of 2017, the defendant Tian, while working in the Cadastral Unit of Luocheng Branch of Luohe City Land and Resources Bureau, used his position responsibility for cadastral registration to provide Zhao with a blank cadastral registration form, forged a false cadastral file, and accepted Zhao a certain amount of cash of 25,000 yuan for personal consumption.

Case example of Law Article 185：
Crimeof acceptance of bribes

Basic Encoder:

From Jan. 2006 to Mar.2007, the defendant Gong used the position of signing and clearing the contract for the coal unloading business of the plant's steam coal during his service as a company manager to obtain benefits for a loading and unloading team. And he illegally accepted the benefit fee of 40,000 yuan from the legal person of the loading and unloading team……

Fact Re-encoder:

From Jan. 2006 to Mar.2007, the defendant Gong used the position of signing and clearing the contract for the coal unloading business of the plant's steam coal during his service as a company manager to obtain benefits for a loading and unloading team. And he illegally accepted the benefit fee of 40,000 yuan from the legal person of the loading and unloading team……

Case example of Law Article 163：
Bribery crime of non-state emplotees

Figure 5: The attention visualization on case examples for Article 185 and Article 163.

word like *"use position" "accept benefit" "accept ... cash"*, etc. Therefore, when using just the representation of basic encoder to predict acceptable law articles, charges and terms of penalty, these two cases tend to be misjudged. As we mentioned in Sec. 4.3, with the distinction vector, our fact re-encoder focuses on extracting distinguishable features like defendants' identity information (e.g., *"company manager" "working in the Cadastral Unit of Luocheng Branch of Luohe City Land and Resources Bureau"* in our examples), which effectively distinguish the applicable law articles and charges of these two cases.

6 Conclusion

In this paper, we present an end-to-end model, LADAN, to solve the issue of confusing charges in LJP. In LADAN, a novel attention mechanism is proposed to extract the key features for distinguishing confusing law articles attentively. Our attention mechanism not only considers the interaction between fact description and law articles but also the differences among similar law articles, which are effectively extracted by a graph neural network GDL proposed in this paper. The experimental results on real-world datasets show that our LADAN raises the F1-score of state-of-the-art by up to 5.79%. In the future, we plan to study complicated situations such as a law case with multiple defendants and charges.

Acknowledgments

References

Stephen Bonner, Ibad Kureshi, John Brennan, Georgios Theodoropoulos, Andrew Stephen McGough, and Boguslaw Obara. 2019. Exploring the semantic content of unsupervised graph embeddings: an empirical study. *Data Science and Engineering*, 4(3):269–289.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *ICML*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.

Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Few-shot charge prediction with discriminative legal attributes. In *COLING*.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *In ICML*.

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.

Benjamin E Lauderdale and Tom S Clark. 2012. The supreme court's many median justices. *American Political Science Review*, 106(4):847–866.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*.

Wan-Chen Lin, Tsung-Ting Kuo, Tung-Jia Chang, Chueh-An Yen, Chao-Ju Chen, and Shou-de Lin. 2012. Exploiting machine learning models for chinese legal documents labeling, case classification, and sentencing prediction. *Processdings of RO-CLING*.

Chao-Lin Liu, Cheng-Tsung Chang, and Jim-How Ho. 2004. Case instance generation and refinement for case-based criminal summary judgments in chinese.

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to predict charges for criminal cases with legal basis. *arXiv preprint arXiv:1707.09168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.

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

Octavia-Maria Sulea, Marcos Zampieri, Shervin Malmasi, Mihaela Vela, Liviu P Dinu, and Josef van Genabith. 2017. Exploring the use of text classification in the legal domain. *arXiv preprint arXiv:1710.09306*.

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

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

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*.

Nuo Xu, Pinghui Wang, Long Chen, Jing Tao, and Junzhou Zhao. 2019. Mr-gnn: Multi-resolution and dual graph neural network for predicting structured entity interactions. *In IJCAI*.

Wenmian Yang, Weijia Jia, XIaojie Zhou, and Yutao Luo. 2019. Legal judgment prediction via multi-perspective bi-feedback network. *arXiv preprint arXiv:1905.03969*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL*.

Haoxi Zhong, Guo Zhipeng, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *EMNLP*.