An Element-aware Multi-representation Model for Law Article Prediction

Huilin Zhong[1], **Junsheng Zhou**[*1], **Weiguang Qu**[1], **Yunfei Long**[2], and **Yanhui Gu**[1]
[1] School of Computer and Electronic Information, Nanjing Normal University, China
[2] School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK CO2 8JT
{zhoujs, wgqu}@njnu.edu.cn, yanhgu@gmail.com
yl20051@essex.ac.uk, elaine1027zhl@hotmail.com

Abstract

Existing works have proved that using law articles as external knowledge can improve the performance of the Legal Judgment Prediction. However, they do not fully use law article information and most of the current work is only for single label samples. In this paper, we propose a Law Article Element-aware Multi-representation Model (LEMM), which can make full use of law article information and can be used for multi-label samples. The model uses the labeled elements of law articles to extract fact description features from multiple angles. It generates multiple representations of a fact for classification. Every label has a law-aware fact representation to encode more information. To capture the dependencies between law articles, the model also introduces a self-attention mechanism between multiple representations. Compared with baseline models like TopJudge, this model improves the accuracy of 5.84%, the macro F1 of 6.42%, and the micro F1 of 4.28%.

1 Introduction

Legal Judgment Prediction(LJP) aims to predict a law case's judgment results given a fact description text. LJP mainly contains three sub-tasks, law article prediction, charge prediction, and terms of penalty prediction. In the civil law system, the correct prediction of law article prediction can help improve the accuracy of charge prediction(Luo et al., 2017). The investigation of law article prediction has significant meaning for LJP.

The law article prediction aims to predict the case's relevant law articles given the fact description (hereinafter abbreviated fact) of a case. In the law article prediction, law articles play an essential role as external information. Luo et al. (2017) uses some candidate law articles to improve the performance of the charge prediction task. However, current researches have two main limitations. One is that certain law articles are considerably similar which makes them difficult to distinguish. Using the representation of overall law articles to extract fact information is not intuitive enough. Another one is that most of the works (Zhong et al., 2018a; Yang et al., 2019; Liu et al., 2019) only predict on single label examples. Meanwhile, in the actual judgment, many cases contain multiple relevant law articles(Zhong et al., 2018b).

Human judge process mainly compares the elements of law article with the case description(Hu et al., 2018), such as the subject of crime (person or specific identity), the object of the crime (person or thing), the purpose and motive of the crime, the harmful behavior, the adverse result, and the crime scene (time or place).

To make full use of the law article information and reduce the confusion in distinguishing different law articles, we have designed a Law Article Element-aware Multi-representation Model (LEMM). LEMM is more related to human cognitive logic and more intuitive based on the law element. We call it LEMM because it extracts fact features specifically by using law article elements and generates multiple law-aware fact representations. Each label has a particular fact representation in classification, which benefits the law article prediction task. Using one vector to distinguish correct law article is inappropriate because the number of relevant law articles is more than 100. Considering law-aware fact representation takes law article as an individual unit, and there are some dependencies between law articles, we capture the relationship between them via the self-attention mechanism. Our LEMM model makes an excellent performance in all evaluation indicators.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6663–6668,
November 16–20, 2020.

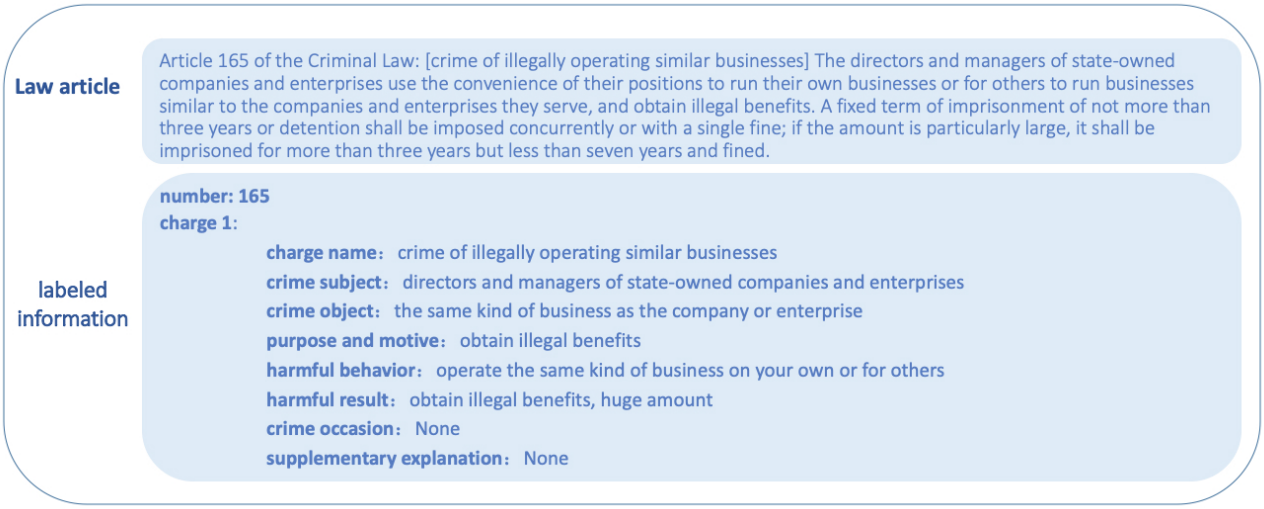

Figure 1: Labeled Law Article

2 Structurally Labeling the Law Articles

Judging whether the law article and case are relevant mainly depends on whether the key elements (including the subject, object, purpose, motive, the harmful behavior, the result of the harm, and the circumstances of the crime) are consistent with the law. Therefore, we divide the law articles into seven elements: 1. the crime subject, 2. the crime object, 3. the purpose and motive of the crime, 4. the harmful behavior, 5. the harmful result, 6. the crime occasion and 7. the supplementary explanation. Since a law article may contain multiple crimes, such law article has multiple groups of elements which correspond to different crimes. As shown in Figure 1, we first divide the content of the law according to the crime and then label the various elements. For elements that are not specified or restricted, we mark them as None. We label 183 candidate law articles of the CAIL dataset (Xiao et al., 2018), which contains a total of 202 crimes.

3 LEMM Model

The fact is a word sequence $\{w_1, w_2, \ldots, w_m\}$. The model uses labeled law articles to help extract features of the fact. The labeled law articles contain the name of crime and the elements of the crime. The name of crime is a word sequence: $\{w_1, w_2, \ldots, w_n\}$. The elements of crime contain seven word sequences: $\{ele_1, ele_2, \ldots, ele_7\}$, where ele_i is $\{w_1, w_2, \ldots, w_{ik}\}$.

Our model contains five components:

Encoder: encode law article elements and fact.

Feature Extraction: use element representations to extract word-level and document-level fact representation by attention mechanism.

Fusion: fuse the word-level and document-level fact representation to law-aware representations.

Relation Extraction: extract the dependencies between law articles by self-attention.

Classification: classify whether the law article is relevant.

3.1 Encoder

The Encoder component contains two encoders, which are element encoder and fact encoder.

3.1.1 Element Encoder

Element Encoder uses BiGRU (Cho et al., 2014) to encoder crime name and crime elements. It takes the hidden state of the last token as the representation of the input. This process is shown as below:

$$ch = BiGRU_{crime}(\{w_1, w_2, \ldots, w_n\}) \quad (1)$$

$$ele_i = BiGRU_i(\{w_1, w_2, \ldots, w_{ik}\}) \quad (2)$$

3.1.2 Fact Encoder

Fact Encoder also uses BiGRU. It takes the hidden state of the last token as document level representation of the fact F and each hidden state as corresponding word representation x_i.

$$F = \{\overleftarrow{h_0}; \overrightarrow{h_m}\} \quad (3)$$

$$x_i = \{\overleftarrow{h_i}; \overrightarrow{h_i}\} \quad (4)$$

$$\overrightarrow{h_i}, \overleftarrow{h_i} = \overrightarrow{GRU}(\overrightarrow{h_{i-1}}, e_i), \overleftarrow{GRU}(\overleftarrow{h_{i+1}}, e_i) \quad (5)$$

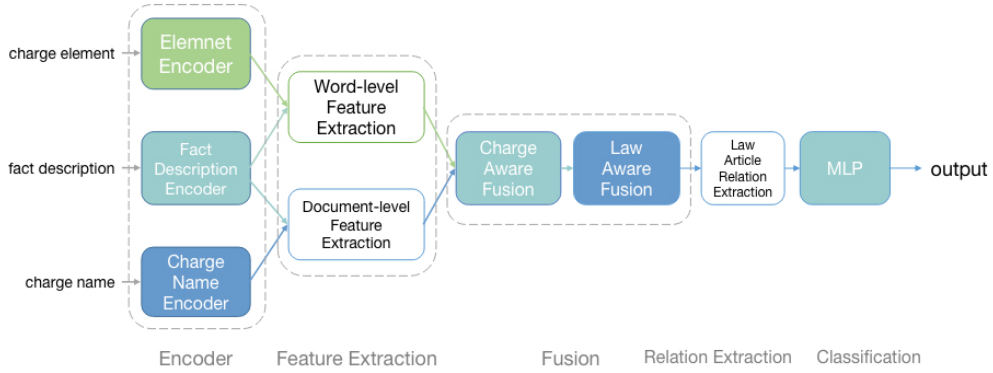

Figure 2: Overview of our LEMM for law article prediction

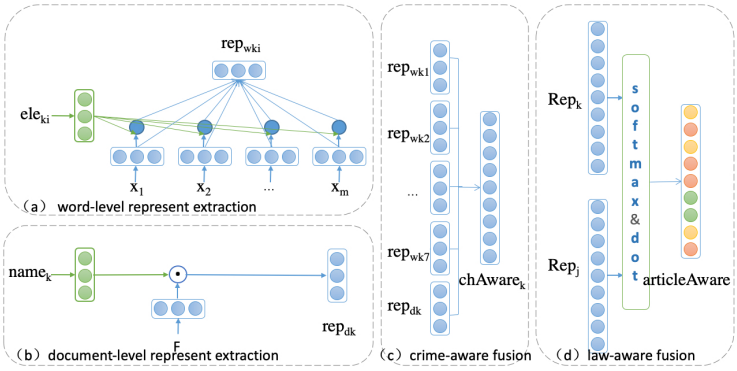

Figure 3: The components of LEMM for law article prediction

3.2 Feature Extraction

Different from Luo et al. (2017) which uses fact information to extract features of law article, we use law elements to extract features of fact and generate multiple representations for one fact. Feature Extraction contains word-level features and document-level features.

3.2.1 Word-level Feature Extraction

We use each law article element as a query to generate word-level representations of the fact by attention mechanism. The calculation is shown as below, where rep_{wi} is the word-level representation of fact extracted by ele_i and f is a non-linear function.

$$\alpha_{ij} = \frac{exp(f_{elei}(ele_i^T) f_x(x_j))}{\sum_{k=1}^{m} exp(f_{elei}(ele_i^T) f_x(x_k))} \quad (6)$$

$$rep_{wi} = \sum_{j=1}^{m} \alpha_{ij} x_j \quad (7)$$

3.2.2 Document-level Feature Extraction

To further strengthen the interaction between the fact and the law articles, we also use crime name representations to extract the document-level features of the fact rep_d via element-wise product.

$$rep_d = f_{ch}(ch) \cdot f_F(F) \quad (8)$$

3.3 Fusion

The Fusion is used to fuse word-level representation and document-level representation. Considering the word-level representations are based on the crime element, the document-level representations are affected by crime name, and crime belongs to law article, we do the crime-level Fusion firstly and then do the law article-level Fusion.

3.3.1 Crime-aware Fusion

We use linear fusion to fuse crime name and the seven elements corresponding to the crime. The document-level case description representation generated by the crime name and the word-level case description representation generated by the elements of the crime are concatenated and put into a linear function for fusion.

$$chAware = f([rep_d; rep_{w1}; \ldots; rep_{w7}]) \quad (9)$$

f is a linear function and $[;]$ means concatenate. $chAware$ is crime-aware representation.

3.3.2 Law-aware Fusion

The Law-aware Fusion is to fuse crime-aware representation based on a law article unit. Some of the law articles only contain one crime, so that we take the crime-aware representation as the article-aware representation.

$$articleAware_s = chAware_u \tag{10}$$

$articleAware$ is the fact representation generated by k-th law article and $chAware_u$ is the fact representation generated by u-th crime. The $crime_u$ belongs to $lawarticle_s$ and the $lawarticle_s$ only has one crime u in content.

When multiple crimes occur in one law article, we hope to select the prominent features of crime-aware presentation. Considering that $argmax$ will cause for failing to return gradients, we use $softmax$ instead.

$$s_{mvt} = \frac{exp(chAware_{vt})}{\sum_{i \in m} exp(chAware_{it})} \tag{11}$$

$$articleAware_m = \sum_{i \in m} s_i \cdot chAware_i \tag{12}$$

$chAware_{vt}$ is the t-th position of the v-th crime-aware representation vector. s_{mvt} is the $softmax$ score of the t-th position of the v-th crime-aware representation.

3.4 Relation Extraction

Considering the entire process from the Encoder to the Feature Extraction, and then to the Fusion, each law article is regarded as an independent individual. So far we have not taken the interaction between law articles into consideration. To extract the interaction between law articles , we use the self-attention mechanism (Vaswani et al., 2017) to calculate the interaction between them.

$$q_i, k_i, v_i = W^T_{(q,k,v)} articleAware_i + b_{(q,k,v)} \tag{13}$$

$$\beta_{ij} = \frac{exp(q_j^T k_i)}{\sum_{n=1}^{|k|} exp(q_j^T k_n)} \tag{14}$$

$$input_i = \sum_{j=1}^{|k|} \beta_i v_j \tag{15}$$

$input_i$ is the new fact representation used to discriminate whether the i-th law article is relevant.

3.5 Classification

We have generated multiple article-aware representations for one fact, and each representation $input_i$ corresponds to a law article. We will use these representations to make classification respectively. Each label has a vector for prediction, which helps to retain more feature information. Unlike other multi-label classifications, where a threshold selects the $softmax$ output results, we use multiple binary classifications.

$$out_i = sigmoid(MLP(input_i)) \tag{16}$$

MLP is a multi-layer perceptron.

4 Experiments

This part includes data selection, experimental parameter setting, baseline model, and detailed experimental results.

4.1 Dataset and Evaluation

We use CAIL 2018 small dataset (Xiao et al., 2018). CAIL(Chinese AI and Law Challenge) is a criminal case dataset for competition released by the Supreme People's Court of China. The details of CAIL can be found in Xiao et al. (2018). Considering the serious long-tail distribution of the sample in the dataset, we only select the samples with more than 300 occurrences of the relevant law. To study the model's performance on low-frequency samples, we also conducted experiments on the complete small dataset.

We use the correct rate, micro/macro accuracy, precision, recall, and F1 as evaluation indicators.

4.2 Experimental Parameter Setting

We use the $Thulac$ (Li and Sun, 2009) tool to segment words, and use $CBOW$ (Rong, 2014) to train word vector on the training data and law article content. The dimension of the word vector is 300. Due to the enormous length of the fact, we only keep the first 256 words of fact. The hidden size is 512. The optimizer is Adam, and the learning rate is 2e-4.

4.3 Experimental Results

We compared our model with LSTM(Cheng et al., 2016), BiLSTM, CNN(Kim, 2014), and the current state-of-the-art TopJudge model. The hidden size is 512, the max word length is 256, the kernel size is [3, 3, 3], and the pooling size is [3, 3, 3].

Model	Acc	Macro			Micro		
		P	R	F1	P	R	F1
TopJudge	71.41	81.20	73.88	76.04	80.31	79.40	79.85
CNN	71.36	78.60	73.88	75.54	79.32	78.89	79.10
LSTM	72.08	80.51	76.58	77.66	79.87	80.42	80.14
BiLSTM	72.27	79.45	78.01	78.07	78.59	81.81	80.17
LEMM	**77.25**	**83.91**	**82.11**	**82.46**	**83.73**	**84.55**	**84.13**

Table 1: Results on CAIL 2018 small (filtered)

Model	Acc	Macro			Micro		
		P	R	F1	P	R	F1
TopJudge	65.45	61.11	46.65	50.17	78.13	72.47	75.19
CNN	67.18	62.55	51.02	53.91	77.66	75.20	76.41
LSTM	68.99	64.52	56.71	58.57	77.05	78.26	77.65
BiLSTM	70.32	65.07	59.63	60.51	76.73	80.53	78.58
LEMM	**72.13**	**73.53**	**61.21**	**64.69**	**81.47**	**81.65**	**81.56**

Table 2: Results on CAIL 2018 small (whole)

We tested our model on the complete and filtered CAIL small dataset. The experimental results are shown in Tabel 1 and Tabel 2. The experiment results show:

(1) Our model has achieved outstanding performance in all evaluation indicators. Compared with TopJudge, our model has achieved 12.42% and 3.34% improvement in macro accuracy and micro accuracy respectively, and 14.56% and 9.18% improvement in macro recall and micro recall respectively.

(2) The performance of TopJudge(current state-of-the-art model) on the two datasets is worse than that of LSTM and BiLSTM. Base on the result, we suspect that joint learning of TopJudge's three subtasks causes more error propagation, and terms of penalty prediction is greatly affected by external factors.

4.4 Ablation Experiment

We compared the LEMM model with some variant models on the screened dataset. The experimental results are shown in Tabel 3. -R means to remove the law article relationship extraction module. The model fact-art puts the entire word sequence of the law article into BiGRU for encoding and use the law article representation to extract features of the fact.

Model	Acc	Macro			Micro		
		P	R	F1	P	R	F1
LEMM	**77.25**	**83.91**	**82.11**	**82.46**	**83.73**	**84.55**	**84.13**
- R	75.85	85.13	78.77	81.18	84.48	82.93	83.70
fact-art	72.68	83.05	79.66	80.62	82.06	82.88	82.47

Table 3: Ablation Experiment Results

The ablation experiment shows that the law article relationship significantly contributes to the improvement of the accuracy rate and recall rate. Nevertheless, the precision of the model has been slightly dropped with law article relationship. There might be some noise information in extracting the relationships, which affects the accuracy of the model.

The performance has a sharp drop without manual labeling law article elements. This verifies the labeled law article information is useful in extracting facts.

5 Conclusion

We propose a model that predicts relevant law articles on multi-label samples by simulating the human judging process. Our proposed LEMM model uses elements of the manually labeled law articles to generate multiple representations of a fact. It uses self-attention to capture dependencies between law articles and makes a unique representation for each candidate label for prediction. The experiments verify that the element-aware multi-representation can better extract features of the factual information and the dependencies between law articles are beneficial to the law article prediction task. The model achieves state-of-the-art performance in benchmark datasets. It also fills the gap between experimental and practical applications on multi-label samples.

Acknowledgments

We thank all reviewers for the valuable comments. This work is supported by the National Natural Science Foundation of China (No. 61472191 and No. 61772278).

References

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *CoRR*, abs/1601.06733.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Zhongguo Li and Maosong Sun. 2009. Punctuation as implicit annotations for chinese word segmentation. *Computational Linguistics*.

Zonglin Liu, Meishan Zhang, Ranran Zhen, Zuoquan Gong, Nan Yu, and Guohong Fu. 2019. Multi-task learning model for legal judgment predictions with charge keywords. *Journal of Tsinghua University(Science and Technology)*, 59(7):497.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Xin Rong. 2014. word2vec parameter learning explained. *CoRR*, abs/1411.2738.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

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

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. AAAI Press.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018a. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549, Brussels, Belgium. Association for Computational Linguistics.

Haoxi Zhong, Chaojun Xiao, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2018b. Overview of CAIL2018: legal judgment prediction competition. *CoRR*, abs/1810.05851.